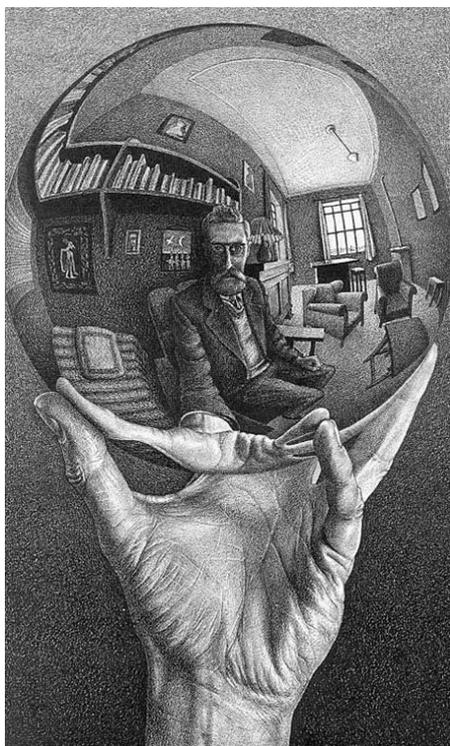
LORENZO PANTIERI & TOMMASO GORDINI

L'ARTE DI SCRIVERE CON LATEX

LORENZO PANTIERI & TOMMASO GORDINI

L'ARTE DI SCRIVERE CON LATEX

NUOVA EDIZIONE
INTERAMENTE RIVEDUTA

Prefazione di Enrico Gregorio

Gruppo Utilizzatori Italiani di TEX e LATEX
Febbraio 2012

Lorenzo Pantieri · Tommaso Gordini
L'arte di scrivere con LaTeX

COLOPHON
Questo lavoro è stato realizzato con LaTeX su Mac usando ArsClassica, una rielaborazione dello stile ClassicThesis di André Miede ispirato a *Gli elementi dello stile tipografico* di Robert Bringhurst.

La copertina riproduce l'incisione *Tassellazione del piano con uccelli* di Maurits Cornelis Escher.

Il frontespizio riproduce la medesima incisione presente sulla copertina e la litografia *Mano con sfera riflettente*, dello stesso autore (tutte le riproduzioni delle opere di Escher contenute nella guida sono tratte da ☞ ESCHER).

CONTATTI
☞ GUIT · Sito ufficiale del GuIT
☞ lorenzo.pantieri@iperbole.bologna.it · Scrivi a Lorenzo Pantieri
☞ illinguista1972@gmail.com · Scrivi a Tommaso Gordini

La citazione è un utile sostituto dell'arguzia.

— Oscar Wilde

Dedicato a tutti gli appassionati di LaTeX.

INDICE

ELENCO DELLE FIGURE

ELENCO DELLE TABELLE

PREFAZIONE

La storia di TeX in Italia è lunga: secondo i resoconti del TeX Users Group, il primo convegno su TeX tenuto in Europa è stato a Como nel maggio 1985 e ne esiste perfino un volume di rendiconti a cura di Dario Lucarella.

Per lunghi anni, però, è mancato un manuale in italiano che spiegasse ai possibili utenti di LaTeX come cominciare a scrivere documenti con questo sistema. Da pochi anni è disponibile la traduzione italiana della *Not So Short Introduction to LaTeX 2ε*, che però è ferma a una versione abbastanza antiquata. Si trovano nei meandri di Internet altre guide, alcune scritte fin dall'inizio in italiano, altre tradotte. Tutte, compresa la *Not So Short*, soffrono di un grave difetto: rispecchiano in gran parte il modo in cui il loro autore ha imparato TeX e LaTeX.

Ogni utente di LaTeX ha sviluppato le proprie tecniche, raccolte qua e là in modo spesso disordinato, e difficilmente si adatta a cambiare anche se ne scopre di nuove e anche più efficienti. Chiunque abbia provato a raccogliere documenti scritti da diverse persone se ne rende conto a una prima occhiata: molti di questi documenti hanno trovato posto nelle raccolte di *orrori*. Sia chiaro, nessuno è immune dagli *orrori*, compreso chi scrive: quando penso a come scrivevo codice LaTeX anni fa mi vengono i brividi. Quando leggo una guida a LaTeX la prima cosa che faccio è cercare gli *orrori* e inevitabilmente li trovo, così come li si trova molto facilmente nei preamboli che circolano fra chi scrive tesi di laurea o di dottorato.

Non vorrei dare l'impressione di criticare gente che ha speso molto tempo per mettere a disposizione di tutti le proprie conoscenze: ogni guida ha pregi e difetti, solo che i difetti hanno la curiosa tendenza di diffondersi più dei pregi. Anche nel mondo LaTeX evidentemente vale il principio che la moneta cattiva scaccia quella buona.

Che dire di questa guida scritta da Lorenzo Pantieri? Che forse è l'unica che affronta alla radice il problema descritto. Ciò che a prima impressione può apparire pignoleria è invece profonda curiosità di andare alla ricerca del modo migliore per risolvere i problemi. Mi riferisco alle innumerevoli domande poste da Lorenzo sul forum del GuIT su questioni stilistiche, su come affrontare problemi tipografici o più strettamente riguardanti LaTeX. Questa curiosità non è pedanteria fine a sé stessa: è desiderio di esplorare il linguaggio LaTeX per trovare sempre la soluzione più efficace.

Il risultato è una guida molto piacevole da leggere, che introduce al linguaggio LaTeX in modo chiaro evitando per quanto possibile le complicazioni che spesso si trovano in altri scritti simili. Certo, non è completa: se pensiamo che il *LaTeX Companion* ha più di mille pagine, siamo molto distanti da questo obiettivo, che lo stesso autore dice di non avere. Nessuna guida potrà mai essere completa, visto che LaTeX è usato per comporre documenti di generi diversissimi, dalla matematica alla fisica, al diritto o alle edizioni critiche di testi classici. In una guida introduttiva, soprattutto a LaTeX, è importante la chiarezza nella spiegazione dei motivi che impongono all'utente certe scelte, con l'indicazione di dove trovare le soluzioni ai problemi che nella guida stessa sono solo accennati.

Devo menzionare l'eccellente guida scritta con grande competenza da Claudio Beccari, che però è rivolta a un pubblico diverso: sicuramente chi avrà letto *L'arte di scrivere con LaTeX* sarà in grado, se lo desidera, di affrontare gli argomenti più complessi oggetto di quell'opera.

Questa è solo la prima edizione: possiamo essere sicuri che Lorenzo sarà sempre pronto ad aggiunte e correzioni in modo che chi la legge trovi facilmente la risposta ai propri problemi. Sarà un piacere provare a dare una mano sul forum in modo che questa guida diventi ancora migliore e conquisti sempre nuovi lettori contribuendo alla diffusione di TEX e LATEX in Italia.

Padova, 30 marzo 2008 Enrico Gregorio

SU QUESTA NUOVA EDIZIONE

La seconda edizione dell'*Arte* ha la grande novità di un secondo autore, Tommaso Gordini, la cui pignoleria linguistica è pari alla pignoleria TEXnica di Lorenzo Pantieri. Il risultato è una profonda revisione del testo: anche grazie ai consigli e suggerimenti dei numerosi lettori, questa guida è arrivata a una maturità che la rende indispensabile a chi voglia imparare a maneggiare LATEX con facilità. Gli autori perdoneranno la parola "pignoleria" usata con grande affetto: è caparbietà e ostinazione nella ricerca del modo migliore di trasmettere i concetti al lettore, evitando come la peste i tecnicismi e le parole arcane tanto frequenti nei manuali di software. Credo che l'obiettivo sia stato raggiunto con successo.

Resta solo da augurare agli autori un buon riposo in vista della terza edizione. Il mondo TEX è sempre in fermento: XƎTEX e LuaTEX promettono nuovi traguardi e possibilità per chi ama comporre i propri testi secondo i canoni dell'arte tipografica tramandata da secoli.

Padova, 16 luglio 2010 Enrico Gregorio

SULLA REVISIONE

Troppi e troppo consistenti sono i cambiamenti di questa nuova revisione per poterne rendere conto qui compiutamente: al lettore il compito di scoprirli.

Basterà ricordare che la guida è stata completamente riscritta: alleggerita di numerosi argomenti che poco avrebbero giovato all'utente inesperto e moderatamente riappesantita con altri ritenuti più utili. Sostanzialmente invariato rispetto alle edizioni precedenti, anche se qua e là profondamente rimaneggiato per una maggiore chiarezza, rimane invece l'impianto complessivo dell'opera.

Non è la terza edizione preconizzata un anno addietro dal nostro Egregio prefatore, questa, ma crediamo di meritarci ugualmente il riposo augurato. A tutti il nostro più sonoro «buona lettura!».

Cesena-Padova, 9 settembre 2011 Gli Autori

RINGRAZIAMENTI

Desidero ringraziare innanzitutto i membri dello staff del GuIt e poi tutti quelli che hanno discusso con me sul forum del Gruppo, in particolare Fabiano Busdraghi, Gustavo Cevolani, Rosaria D'Addazio, Massimiliano Dominici, Gloria Faccanoni, Gianluca Gorni, Maurizio Himmelmann, Jerónimo Leal, Lapo Filippo Mori, Andrea Tonelli, Emiliano Giovanni Vavassori ed Emanuele Vicentini, per l'insostituibile aiuto fornito nella redazione di questo lavoro, le spiegazioni dettagliate, la pazienza e la precisione nei suggerimenti, le soluzioni fornite, la competenza e la disponibilità: grazie mille, ragazzi!

Rivolgo un ringraziamento davvero particolare al professor Enrico Gregorio, per i suoi impagabili insegnamenti e per avermi concesso l'onore di scrivere la prefazione alla guida.

Un "grazie" altrettanto speciale va al professor Claudio Beccari, per i consigli durante la revisione di un'opera che senza la sua pazienza non avrebbe mai raggiunto la forma attuale.

Ringrazio infine André Miede, per aver realizzato lo stile ClassicThesis (con cui è composto questo lavoro), *Daniel Gottschlag*, che mi ha dato lo spunto per realizzarne un'originale rielaborazione, e Ivan Valbusa, per i consigli grafici, le idee e per aver creato lo stile con cui è composta la bibliografia di questa guida.

Lorenzo Pantieri

Un libro come questo non può *mai* dirsi veramente concluso, e per fortuna il vuoto lasciato dalla fine del lavoro non è così grande. Ai ringraziamenti di Lorenzo, che faccio miei, dovrei aggiungere quelli a un numero imprecisato di persone: lo faccio silenziosamente, e so che non me ne vorranno per non vedere il proprio nome in queste righe.

Voglio però nominare e ringraziare, non necessariamente in quest'ordine, le quattro impigliatesi più saldamente nella rete.

I miei genitori: devo a loro il fatto di capirci (qualche volta) qualcosa. Direi che hanno fatto un buon lavoro.

Alberto, che per primo mi disse: «infatti non è Word, è *làtek*!». Non nascondo che lì per lì l'osservazione mi lasciò del tutto indifferente.

Enrico, il saggio Vecchio Lupo che ha una pazienza grande come il sole e che (ancora) non perde il pelo. Il resto lo conosce.

Eleonora, adorabile e impotente compagna in quest'avventura: senza di lei non potreste leggere quello che state leggendo. Questo libro è più suo che mio.

Tommaso Gordini

Cesena-Padova, 16 luglio 2010

INTRODUZIONE

Abbiamo visto che la programmazione è un'arte,
perché richiede conoscenza, applicazione, abilità e ingegno,
ma soprattutto per la bellezza degli oggetti che produce.

— Donald Ervin Knuth

LaTeX è un programma di composizione tipografica liberamente disponibile, indicato soprattutto nella preparazione di documenti scientifici ai più elevati livelli di qualità. Lo scopo di questo lavoro, rivolto sia a chi muove i primi passi in LaTeX sia a quanti già lo conoscono, è di offrire ai suoi utenti italiani le conoscenze essenziali per poterlo usare con successo.

I concetti fondamentali della materia, raccolti da svariati manuali, vengono presentati nel modo più chiaro e organico possibile; nel contempo si fornisce un vasto campionario di esempi e si analizzano alcuni problemi tipici della stesura di una pubblicazione scientifica o professionale in italiano, indicando per ciascuno le soluzioni per noi migliori.

Queste derivano principalmente dalle numerose discussioni presenti sul forum del GuIt (Gruppo Utilizzatori Italiani di TeX e LaTeX, ☞ GUIT), che resta sempre un eccellente riferimento per tutti i temi trattati in questa guida, la cui lettura, lo ricordiamo, presuppone una familiarità almeno basilare con l'uso del calcolatore e di Internet.

In linea generale si è seguita la prassi di non scandagliare troppo i vari argomenti: dei pacchetti citati (ma non di tutti), infatti, si analizzano soltanto le impostazioni più importanti e se ne suggerisce l'uso, indirizzando alla relativa documentazione chi voglia approfondirne la conoscenza.

L'esposizione degli argomenti è articolata come segue.

IL PRIMO CAPITOLO traccia una breve storia di LaTeX, indicandone idee di fondo e peculiarità.

IL SECONDO CAPITOLO spiega come installare e aggiornare LaTeX sul proprio calcolatore.

IL TERZO CAPITOLO tratta alcuni aspetti generali indispensabili per comprendere il funzionamento del programma.

IL QUARTO CAPITOLO descrive gli strumenti per trattare il testo.

IL QUINTO CAPITOLO esplora uno dei principali punti di forza di LaTeX: la composizione di formule matematiche.

IL SESTO CAPITOLO presenta i concetti e gli strumenti essenziali per comporre le tabelle, includere le figure in un documento e gestire la collocazione degli oggetti mobili sulla pagina.

IL SETTIMO CAPITOLO presenta gli strumenti per realizzare e gestire la bibliografia.

L'OTTAVO CAPITOLO illustra le nozioni essenziali per generare l'indice analitico.

IL NONO CAPITOLO espone alcuni suggerimenti per fare in modo che LaTeX produca risultati diversi da quelli predefiniti.

IL DECIMO CAPITOLO dà alcuni suggerimenti per migliorare l'impaginazione del documento.

L'APPENDICE A descrive sinteticamente le principali norme tipografiche per l'italiano.

L'APPENDICE B presenta gli strumenti per scrivere in latino e in greco.

L'APPENDICE C, infine, fornisce gli elementi essenziali per realizzare una videopresentazione.

Questo *non* è un manuale su LaTeX, ma piuttosto un tentativo di riordinare in forma scritta appunti accumulatisi nel tempo, via via che divenivamo abituali utenti di questo programma. In qualità di semplici appassionati non abbiamo nulla da insegnare; d'altra parte abbiamo studiato LaTeX e l'abbiamo usato intensamente, acquisendo una certa esperienza che ci piacerebbe condividere con altri utenti.

Se avete idee su argomenti da aggiungere, togliere o modificare in questo libro, o se vi dovesse capitare di notare errori, sia di battitura che di sostanza (ed è probabile che ce ne siano parecchi, e del primo e del secondo tipo), fareste davvero una cosa gradita comunicandocelo, così da poterli correggere nelle versioni successive del lavoro. In modo particolare, ci interessano i commenti di chi è alle prime armi su quali parti di questa guida sono di più facile comprensione e quali invece potrebbero essere spiegate meglio.

Ecco lo spirito che ci ha guidati in questo lavoro: speriamo che possiate usare LaTeX con il nostro stesso piacere.

1 STORIA E FILOSOFIA

Questo capitolo presenta una breve storia di TeX e LaTeX, e ne indica idee di fondo e peculiarità.

1.1 STORIA

1.1.1 TeX

TeX è un programma di composizione tipografica realizzato da Donald Ervin Knuth, professore di Informatica all'Università di Stanford (USA), e distribuito con una licenza di software libero.

Nel 1977, Knuth cominciò a scrivere il suo "motore" di tipocomposizione per esplorare le potenzialità degli strumenti di stampa digitale che allora cominciavano a muovere i primi timidi passi in campo editoriale. La qualità tipografica delle proprie pubblicazioni (e soprattutto del capolavoro The Art of Computer Programming, in più volumi ricchi di formule matematiche), constatava Knuth, era in crescente peggioramento: a quel tempo, infatti, gran parte della matematica si componeva con la macchina per scrivere, alzando e abbassando il carrello per indici ed esponenti e cambiando testina per i simboli (si veda la figura 2 nella pagina seguente). Con TeX, il Professore sperava di mettere un freno a questa tendenza.

Figura 1: Donald Ervin Knuth (l'immagine è tratta da ☞ KNUTH).

Il programma ha visto la luce nel 1982, e da allora ha subito costanti aggiornamenti e miglioramenti tra cui quello del 1989, che gli ha permesso di supportare un numero considerevolmente maggiore di caratteri (anche non latini). TeX è rinomato per essere estremamente stabile ed eseguibile su diversi tipi di calcolatore, ed è stato rivisto per l'ultima volta nel 2008. Il suo numero di versione converge a π (attualmente è $3,141\,592\,6$).

The TeXbook, scritto da Donald Knuth, è il manuale d'uso di TeX e uno dei libri più completi su questo programma. Attualmente TeX è un marchio registrato della $\mathcal{AMS}$ (American Mathematical Society).

1.1.2 Etimologia

Knuth ha nascosto un trabocchetto nel nome del programma: TeX, infatti, si pronuncia *tèch* (aspirando il *ch* finale) e non com'è scritto, perché è una

Abitualmente lo sviluppo (8) viene però scritto nella forma

$$f(x) = c_0 + \sum_{n=1}^{\infty} [c_n \cos(nx) + s_n \sin(nx)] \quad (8a)$$

$$= \sum_{n=0}^{\infty} [c_n \cos(nx) + s_n \sin(nx)] \quad (8b)$$

$$c_0 = \frac{1}{2\pi} \int_{-\pi}^{\pi} dx\, f(x), \quad (9a)$$

$$c_n = \frac{1}{\pi} \int_{-\pi}^{\pi} dx\, \cos(nx)\, f(x), \quad (9b)$$

$$s_n = \frac{1}{\pi} \int_{-\pi}^{\pi} dx\, \sin(nx)\, f(x). \quad (9c)$$

Allo sviluppo (8) faremo riferimento come allo sviluppo in serie di Fourier della funzione f(x). Il nome ricorda il matematico francese Jean Baptiste Joseph Fourier (1768-1830), cui è dovuta l'invenzione di questa importantissima tecnica matematica.

Come è chiaro dalle (9), la costante c_0 rappresenta il valor medio della funzione f(x) nell'intervallo $-\pi \le x \le \pi$; le costanti c_n, $n = 1, 2, 3, \ldots$, dipendono solo dalla parte pari di f(x), laddove le s_n, $n = 1, 2, 3, \ldots$, dipendono solo dalla parte dispari di f(x). Infatti se si sostituisce nelle (9) l'identità

$$f(x) = f_p(x) + f_d(x), \quad (10)$$

con

$$f_p(x) = \frac{1}{2}[f(x) + f(-x)], \quad (11a)$$

$$f_d(x) = \frac{1}{2}[f(x) - f(-x)], \quad (11b)$$

identità che implica

$$f_p(x) = f_p(-x), \quad (12a)$$

$$f_d(x) = -f_d(-x), \quad (12b)$$

Figura 2: Pagina di una dispensa di matematica degli anni Settanta del secolo scorso, dattilografata, con aggiunte a mano.

parola greca scritta in greco maiuscolo (in lettere minuscole si scriverebbe τεχ). Si tratta di un'antichissima radice indoeuropea comune non soltanto ai greci τέκτων (pron. *tèkton*, "artefice") e τέχνη (pron. *tèchne*, insieme "arte" e "mestiere"), per esempio, ma viva ancora oggi negli usatissimi *tecnica*, *politecnico*, *architetto* e in numerose altre parole italiane. L'etimologia appena spiegata illumina la scelta di Knuth: TeX sarebbe stato il nome perfetto per un programma che compone documenti "allo stato dell'arte".

Knuth dice che «se TeX è ben pronunciato, lo schermo del calcolatore si appanna leggermente». La lettera *X*, infatti, corrisponde a una forte aspirazione non esistente in italiano, ma presente in molte lingue attualmente parlate sul pianeta: oltre che in greco, si trova nel tedesco *Ba*ch, nello scozzese *Lo*ch, nello spagnolo J*uan* e *Me*x*ico*, nel russo *X*орошо (pron. h*arasciò*, "bene"), nel cinese 你好 (pron. *nǐ* h*ǎo*, "ciao"), solo per fare qualche esempio.

Lo stesso Knuth, però, ammette le diverse pronunce nazionali (che convergono in un *tèk* universale), rassicurando gli utenti che non è certo per il fatto di sentire TeX detto come ognuno preferisce che andrà su tutte le furie. Tuttavia, indica nella Grecia il Paese in cui oggi si può ascoltare la pronuncia più corretta di questa parola.

1.1.3 LaTeX

LaTeX (La(mport)TeX) è un programma di composizione tipografica realizzato da Leslie Lamport e liberamente disponibile, che usa TeX come motore di tipocomposizione.

LaTeX *non* è TeX, però. Per rendere l'idea della differenza tra i due programmi, potremmo paragonare TeX a un corpo, e LaTeX al più popolare degli "abiti" (fatto, però, di istruzioni in linguaggio TeX) che nel corso degli anni gli sono stati confezionati addosso per avvicinarlo al pubblico con sembianze "amichevoli".

LaTeX è stato progettato per automatizzare in una volta sola tutte le più comuni operazioni necessarie a realizzare un documento e, tramite imposta-

Tabella 1: Cronologia di TEX e LATEX.

1977	Knuth comincia a scrivere TEX
1978	Lamport comincia a scrivere il primo nucleo di LATEX
1982	Prima versione pubblica di TEX
1984	Knuth pubblica la prima edizione di *The TEXbook*
1985	Lamport pubblica LATEX 2.09 e la prima edizione di *LATEX. A Document Preparation System*
1994	LATEX 2_ε
2008	Ultima revisione di TEX
2009	Ultima revisione di LATEX

zioni professionali predefinite, permette all'autore di impaginare e stampare il proprio lavoro *ai più elevati livelli di qualità tipografica*.

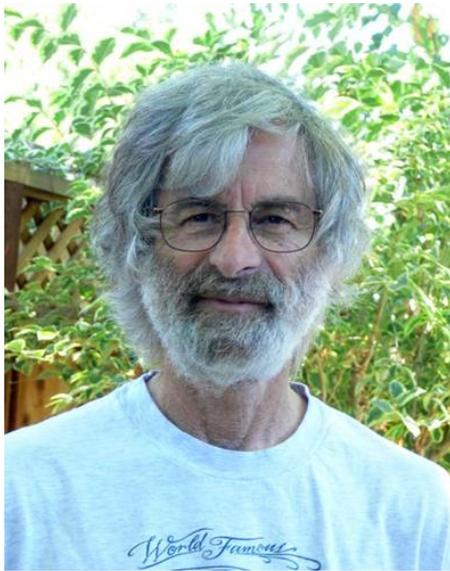



Figura 3: Leslie Lamport (l'immagine è tratta da ☞ LAMPORT).

Lamport, che collaborava con Knuth allo sviluppo di TEX, cominciò a scrivere LATEX alla fine degli anni Settanta del secolo scorso, quando TEX non era ancora stato pubblicato. La prima versione pubblica di LATEX risale al 1985, e da allora il programma è stato continuamente aggiornato e migliorato. Per molti anni il suo numero è rimasto fissato a 2.09 e le successive revisioni sono state identificate con le loro date.

Nel 1994, finalmente, un gruppo di programmatori guidato da Frank Mittelbach lo ha aggiornato in modo sostanziale, includendovi tutte le versioni successive alla 2.09 e numerosi altri miglioramenti. Per distinguerla da quella precedente, la nuova edizione è stata battezzata LATEX 2_ε, e costituisce l'oggetto di questa guida. *LATEX. A Document Preparation System*, scritto da Leslie Lamport, è il manuale ufficiale di LATEX.

Il futuro LATEX3 si profila come un progetto a lungo termine: i costanti aggiornamenti di LATEX, la cui ultima versione pubblica è del 2009, ne costituiscono le tappe di avvicinamento.

1.2 FILOSOFIA

1.2.1 Composizione sincrona e asincrona

La caratteristica che più differenzia LATEX dagli altri elaboratori di testo è il fatto che l'introduzione del testo e la sua composizione avvengono *in tempi diversi* [Beccari, 2011].

Per modificare un documento scritto con un comune *word processor* (come Microsoft Word), l'autore agisce direttamente sul testo già composto così come gli appare sul monitor, e ogni sua azione si traduce in una variazione *immediata* di quel testo. Perciò questo tipo di composizione è detto "composizione sincrona". Per essere *davvero* sincrono e operare con un ritardo

trascurabile tra intervento e visualizzazione, però, il programma deve puntare tutto sulla rapidità della presentazione, il che rende di fatto impossibile ottenere una composizione perfetta, per via della molto più accurata elaborazione del testo di partenza che essa richiede. È anche vero che oggi i programmi di videoscrittura sono estremamente rapidi e che la qualità della loro composizione migliora sensibilmente a ogni nuova versione, ma il compromesso tra velocità e qualità esiste sempre.

La "composizione asincrona", invece, consiste nell'introdurre il testo in un editor concentrandosi unicamente su struttura logica e contenuto del documento (senza perciò preoccuparsi di come apparirà), per darlo "in pasto" a un compositore che lo impagina (LaTeX, in questo caso) solo *successivamente*. L'autore, naturalmente, può modificare in ogni modo il proprio lavoro anche dopo la composizione, ma deve tenere ben presente che LaTeX non si limita ad aggiustarlo *nel punto* in cui è stato modificato e basta, come farebbe un programma qualunque, ma riorganizza sempre *l'intero capoverso* (e, di conseguenza, l'intera pagina) nel migliore dei modi.

Va da sé che questo secondo tipo di composizione è meglio del primo, perché la velocità di visualizzazione passa in secondo piano a tutto vantaggio della qualità: LaTeX elabora il testo sempre *nel suo complesso*, e per questo motivo può fare le scelte d'impaginazione migliori.

1.2.2 Istruzioni di marcatura

L'idea di Lamport era forte. E LaTeX centra in pieno l'obiettivo: l'autore può (quasi *deve*, si direbbe) astrarsi dai dettagli estetici che con un altro elaboratore di testo sarebbe costretto a introdurre a mano, per indirizzare le proprie energie sul *contenuto* del lavoro e sulla scansione delle sue parti.

LaTeX pretende dall'utente considerazioni sul *cosa*: «il mio documento sarà composto da un certo numero di capitoli, ciascuno diviso in paragrafi numerati, avrà indice generale e analitico, delle figure e qualche tabella». E questo è tutto.

Al *come* pensa LaTeX, e lo fa molto bene. Per esempio, uno stesso testo sorgente può generare in teoria documenti radicalmente diversi soltanto cambiandone la classe o caricando un pacchetto che agisce in modo globale su di esso. (Si è detto *in teoria* perché, nella pratica, qualche aggiustamento manuale si rende di fatto *sempre* necessario.)

Un file da comporre con LaTeX è scritto in una "lingua" particolare costituita da *marcatori* (o *etichette logiche*, *mark-up* in inglese), ovvero le istruzioni che il programma deve eseguire per trattare nel modo specificato dall'autore la porzione di testo a cui si riferiscono. Le etichette logiche sono, in una parola, i *comandi* (chiamati anche, nel gergo di LaTeX, *macro*, abbreviazione di *macroistruzioni*) e gli *ambienti*.

Nonostante faccia dei marcatori uno dei propri punti di forza (per cui molte delle tradizionali operazioni sul testo vengono automatizzate), LaTeX definisce anche gli strumenti per regolare finemente (e a mano, questa volta) il risultato durante la revisione finale del documento, passaggio ancora insostituibile. In tipografia, infatti, le variabili sono troppe e troppo variegate per poter essere gestite in una volta sola da un unico programma; e l'autore deve accettarlo, se pretende un prodotto di alta qualità.

Ecco che il file prodotto dall'editor è a tutti gli effetti un codice scritto in un linguaggio di programmazione, dato che contiene sia il testo vero e proprio del documento, sia le istruzioni di marcatura (i comandi) che dicono a LaTeX di comporre quello che gli si dà in pasto secondo lo stile particolare del documento che si è scelto di scrivere.

Non si spaventi chi non conosce bene l'inglese: i comandi di LaTeX (che specificano l'inizio di un capitolo o una particolare modalità di stampare

una formula matematica, per esempio), sono in un inglese molto semplice, e anche là dove ci sono abbreviazioni risultano di solito comprensibilissimi e facilmente memorizzabili.

Un esempio

Per dare l'idea di come appare un documento da comporre con LaTeX, di seguito si riportano alcune righe di *codice sorgente*, cioè l'insieme di testo, numeri, simboli e istruzioni di marcatura da scrivere nell'editor.

```
Due matrici $n\times n$ complesse $A$ e $B$ si dicono \emph{simili}
se esiste una matrice $n\times n$ invertibile $T$ tale che
\begin{equation}
B=T^{-1}AT.
\end{equation}
```

Il codice sorgente viene composto da LaTeX che, attraverso TeX, produce il documento tipocomposto (*typeset*). Se il risultato non soddisfa, non si può modificare direttamente il documento a schermo, ma bisogna correggere il testo sorgente e poi ricomporlo.

L'esempio seguente riporta a sinistra lo stesso codice sorgente dell'esempio appena visto e a destra il risultato della composizione.

```
Due matrici $n \times n$
complesse $A$ e $B$ si dicono
\emph{simili} se esiste una
matrice $n \times n$
invertibile $T$ tale che
\begin{equation}
B=T^{-1}AT.
\end{equation}
```

Due matrici $n \times n$ complesse A e B si dicono *simili* se esiste una matrice $n \times n$ invertibile T tale che

$$B = T^{-1}AT. \tag{1.1}$$

Nei prossimi capitoli si spiegheranno tutte le istruzioni usate nell'esempio. Tuttavia, anche con pochi rudimenti di inglese si capisce facilmente quello che il linguaggio di marcatura ha specificato.

1.2.3 LaTeX: pro e contra

I vantaggi di LaTeX rispetto agli altri elaboratori di testo sono innumerevoli. Di seguito se ne elencano alcuni.

- LaTeX compone documenti al massimo grado di professionalità e presenta caratteristiche di qualità e stabilità sconosciute agli altri elaboratori di testo.
- L'autore pensa a struttura e contenuto del documento, LaTeX si occupa della sua impaginazione.
- LaTeX genera strutture complesse come riferimenti incrociati, indici e bibliografie con grande efficienza e flessibilità.
- LaTeX gestisce in modo impeccabile la composizione tipografica di formule matematiche.
- LaTeX è gratuito, multilingue e multipiattaforma (un suo file sorgente è puro testo, e dunque viene letto tranquillamente su tutti i sistemi operativi; si veda, però, il paragrafo 3.2 a pagina 18).
- LaTeX ha una struttura modulare che permette di estenderne le capacità per eseguire compiti tipografici non direttamente gestiti dal programma.

- LaTeX rende davvero difficile creare documenti mal strutturati e tipograficamente scadenti.
- LaTeX appassiona talmente... che ci si può dimenticare la tazza del caffè sul foglio!

Innegabilmente, LaTeX presenta anche alcuni svantaggi.

- Con LaTeX ci vuole attitudine all'astrazione.
- Con LaTeX la gratificazione non è istantanea, ma ritardata.
- LaTeX «non lavora bene per chi ha venduto la propria anima» [Oetiker *et al.*, 2011].
- Solo gli esperti possono permettersi di uscire dagli stili predefiniti.
- Il fatto che LaTeX raggiunga un'elevatissima qualità soltanto grazie alla composizione asincrona può talvolta rivelarsi un difetto anziché un pregio (almeno se non si considerano i motivi di questo comportamento e non si adatta di conseguenza il proprio metodo di lavoro). LaTeX, infatti, gestisce il capoverso in modo del tutto particolare: rimuoverne anche una sola parola ne determina *sempre* il riallestimento completo. Con risultati a volte indesiderati: il capoverso potrebbe risultare addirittura di una riga più lungo, se LaTeX decidesse che quella è la soluzione migliore, con le ovvie conseguenze se ciò dovesse accadere durante l'ultimissima revisione del documento.
- LaTeX non gestisce pagine più grandi di 33 metri quadrati e non dà denaro alle case di software.

In breve, è opportuno usare LaTeX quando [Mittelbach, Pignalberi e Walden, 2007]:

- l'utente preferisce pensare per strutture logiche;
- i documenti da preparare richiedono coerenza interna;
- i documenti da preparare non hanno un formato completamente definito o potrebbero essere presentati parallelamente in vesti diverse;
- i documenti da preparare contengono molta matematica;
- il materiale è corposo.

Viceversa, si sceglierà un sistema di videoscrittura tradizionale quando:

- l'utente preferisce pensare per strutture visive;
- l'utente non si trova del tutto a proprio agio nel lavorare con un linguaggio di programmazione (un editor per LaTeX aiuta, ma...);
- i documenti da preparare richiedono più flessibilità visuale che consistenza (volantini, biglietti d'invito, dépliant, brochure, eccetera);
- il materiale non è corposo.

In ultima battuta: prima di affidarsi a LaTeX si consiglia caldamente di vagliare *a monte* e con la massima attenzione il tipo di prodotto che si intende realizzare.

1.2.4 Luoghi comuni

Molte delle persone che si avvicinano a LaTeX dopo anni di tormenti alle prese con altri elaboratori di testo si stupiscono quando scoprono che era disponibile da oltre due decenni, pur non avendone mai sentito parlare. Non si tratta certo di una cospirazione ai loro danni, ma solo di «un segreto ben conservato e noto solo a pochi milioni di persone» [Flynn, 2005].

Knuth e Lamport hanno generosamente reso i loro programmi di pubblico dominio fin da subito, e perciò per molto tempo LaTeX è vissuto indisturbato tra le mura delle università senza che nessuno sentisse il bisogno di pubblicizzarlo al di fuori. Oggi, tuttavia, LaTeX è divenuto popolarissimo: innumerevoli editori pubblicano documenti nel suo formato, e centinaia di migliaia di utenti scrivono con esso milioni di documenti ogni giorno.

In questi anni, su LaTeX sono fioriti molti luoghi comuni: per evitare possibili incomprensioni, conviene in questa sede esaminarne i più diffusi.

Leggenda: LaTeX ha solo un font

LaTeX può usare, tra gli altri, *ogni* font di tipo TrueType, OpenType, PostScript e METAFONT. Ciò è più di quanto viene offerto dalla maggior parte degli altri sistemi di composizione tipografica. Il font standard di LaTeX è il Computer Modern, e non il Times New Roman, quindi alcuni restano turbati da un documento diverso dal solito.

Leggenda: LaTeX è un software solo per Unix

Si sente anche dire che LaTeX è un software solo per Unix, o solo per Mac, eccetera. È tutto il contrario: LaTeX funziona sulla maggior parte dei calcolatori oggi sul mercato (supercomputer e palmari compresi): dai PC con Windows ai Mac ai sistemi Unix/Linux. Se LaTeX non gira sul proprio computer, molto probabilmente il sistema operativo installato è solo troppo vecchio.

Leggenda: LaTeX è obsoleto

Proprio il contrario. L'incessante lavoro di migliaia di appassionati in tutto il mondo lo tiene costantemente aggiornato, e quasi ogni giorno si aggiungono agli archivi ufficiali nuove caratteristiche. È dunque indiscutibilmente più aggiornato della maggior parte degli altri sistemi di videoscrittura, e la qualità nel comporre formule matematiche e capoversi è tuttora ineguagliata.

Leggenda: LaTeX non è WYSIWYG

Dipende. Se per WYSIWYG (*What You See Is What You Get*, "ciò che vedi è ciò che ottieni") si intende un software in grado di produrre nel documento finito testo e immagini disposti esattamente come si vede sullo schermo del calcolatore, LaTeX è un programma WYSIWYG della migliore qualità.

Se invece per WYSIWYG si intende un programma di videoscrittura in cui l'autore agisce direttamente sul testo già composto così come gli appare a schermo e ogni sua azione si traduce in un'immediata variazione di quel testo, LaTeX non è WYSIWYG ma WYSIWYM (*What You See Is What You Mean*, "ciò che vedi è ciò che intendi"), perché compone il documento in modo asincrono.

Leggenda: LaTeX è troppo difficile

Questa frase si è sentita dire da fisici in grado di dividere gli atomi, da matematici che sanno dimostrare la trascendenza di π, da uomini d'affari che sanno leggere un foglio di bilancio, da storici che hanno compreso la politica bizantina e da linguisti che sanno decifrare la scrittura lineare B. LaTeX non è immediato come i normali elaboratori di testo, certo, ma quel po' di studio e di pratica iniziali richiesto (possibilmente su documenti elementari) verranno presto ricompensati dalla qualità dei risultati.

Leggenda: LaTeX è solo per matematici e scienziati

Niente affatto. Sebbene sia cresciuto nei campi della matematica e dell'informatica, due delle sue maggiori aree di espansione sono quelle umanistica ed economica, specie da quando ha preso piede l'XML, che ha sollevato nuove esigenze nell'ambito della tipocomposizione automatica.

2 INSTALLARE E AGGIORNARE

Questo capitolo spiega brevemente come procurarsi tutto l'occorrente per usare LaTeX, come installarlo nel proprio calcolatore e come aggiornarne la distribuzione (che in queste pagine si considera *sempre* nell'ultima versione disponibile). I programmi descritti sono gratuiti e, dove non altrimenti indicato, disponibili per tutti e tre i sistemi operativi considerati in questa guida:

- Windows (da XP in poi);
- Mac OS X (da 10.5 Leopard in poi);
- Linux (almeno nelle distribuzioni indicate nel paragrafo 2.4 a pagina 14).

2.1 FERRI DEL MESTIERE

Per creare un documento con LaTeX sono indispensabili almeno tre cose:

1. un editor di testi con cui scrivere il file sorgente;
2. il programma LaTeX, che elabora il file e produce il documento tipocomposto;
3. un programma per visualizzare il documento finito.

Si eviti da subito l'errore, molto comune all'inizio, di confondere LaTeX (un "motore" del tutto invisibile all'utente) con l'editor (ciò che effettivamente appare sullo schermo): i due programmi sono completamente *indipendenti*, tanto che in teoria si può usare un editor qualunque, da quelli più elementari già presenti nel proprio computer a quelli più complessi in grado di gestire numerosi linguaggi di programmazione.

Un altro errore frequente è confondere i due programmi `latex` e `pdflatex`: il primo produce file `.dvi` (oggi indispensabili in circostanze così limitate da non meritare una trattazione in questa guida), il secondo file `.pdf`. Di solito li si confonde perché il nome *LaTeX* viene usato indifferentemente per indicare, e in qualche editor si chiama così anche il pulsante che avvia la composizione.

In questa guida con *LaTeX* si intende *sempre* il programma `pdflatex`, e con le espressioni *documento composto* e *documento finito* un file in formato PDF.

2.1.1 Editor e visualizzatore: TeXworks

L'attuale tendenza degli sviluppatori è di eliminare la necessità di installare programmi diversi creando editor comprensivi di un visualizzatore di PDF interno, e questa è anche la scelta qui consigliata.

Anche se oggi molti software di videoscrittura tradizionali prevedono estensioni per trasformare il testo immesso in codice LaTeX, il consiglio per chi muove i primi passi con questo programma è di usare senz'altro un editor *dedicato*, cioè destinato esclusivamente a LaTeX e ideato per facilitarne il più possibile l'interazione con l'utente: i vantaggi sono molti e sostanziali.

Tabella 2: Percorsi di TEX nei diversi sistemi operativi.

Sistema operativo	Percorso di TEX
Windows	`C:\Programmi\texlive\2011\bin\win32`
Mac OS X	`/usr/texbin`
Linux	`/opt/texbin`

Gli editor che gestiscono codice LATEX sono moltissimi, come si può scoprire con una veloce ricerca in Rete. Tra tutti vanno segnalati almeno:

- TeXShop (☞ TEXSHOP), nativo di Mac OS X;
- Kile (☞ KILE), nativo di Linux;
- Texmaker (☞ TEXMAKER) e TeXstudio (☞ TEXSTUDIO), programmi multipiattaforma con caratteristiche molto simili a quelle di TEXworks, che si descrive nel prossimo paragrafo;
- Emacs (☞ EMACS) nelle sue diverse incarnazioni.

Si ricordi che la scelta finale è anche una questione di gusto.

Un programma molto adatto ai principianti perché permette di evitare facilmente i tipici errori dell'inizio è TEXworks (☞ TEXWORKS), un semplice ambiente di lavoro per LATEX in continua evoluzione, mantenuto da un gruppo di sviluppatori che periodicamente ne rilascia una versione stabile. Per semplicità d'uso e immediatezza è stato scelto per accompagnare TEX Live (si veda il paragrafo 2.1.2 a pagina 12): il programma, perciò, risulterà *già installato* sulla macchina (tranne che su Linux). Si noti che sarà sempre possibile, se lo si desidera, usare l'ultima versione al posto di quella preinstallata.

Caratteristiche del programma

L'interfaccia grafica è piuttosto semplice e intuitiva, facendo a meno dei numerosi pulsanti che di solito affollano le barre degli editor, e le funzioni davvero importanti sono a portata di mano: se ne può vedere un esempio nella figura 4 a pagina 12.

Il programma gestisce numerosi motori di composizione e compone in modo interattivo, segnalando all'autore gli eventuali errori e richiedendo un suo intervento prima di continuare. Offre le normali funzioni di cercare e sostituire, e permette di aggiungere nuovi motori di composizione a quelli che già comprende. Infine, TEXworks è compatibile con la tecnologia SyncTEX: testo sorgente e anteprima cioè, sono *sincronizzati*, e si può "saltare" facilmente da un punto dell'uno al corrispondente punto dell'altro (e viceversa) risparmiando molto tempo. Si noti che attualmente c'è una limitazione, infatti non si può stampare il PDF direttamente dal visualizzatore: si risolve il problema usando un altro programma.

Di seguito si elencano le operazioni fondamentali per il corretto funzionamento di TEXworks:

- nel primo riquadro da sinistra nella parte bassa della finestra del testo sorgente si lasci LF, nel secondo UTF-8;
- si controlli che la prima voce della lista Percorso di TEX e relativi programmi nel pannello Composizione sia quella indicata nella tabella 2;
- si installino i dizionari italiani di OpenOffice.org se si desidera un semplice controllo ortografico.

Commenti speciali e gestione di documenti in più file

Un grande pregio che TeXworks condivide con pochissimi altri editor (solo TeXShop, Emacs e forse Vim permettono qualcosa di simile) è la capacità di interpretare i *commenti speciali*, cioè righe che cominciano con la stringa `% !` da scrivere *come prima cosa* nel testo sorgente e che, come i commenti veri e propri (si veda il paragrafo 3.4.5 a pagina 26), vengono ignorati da LaTeX ma *non* dall'editor, che in modo automatico si autoconfigura corrispondentemente. Non sono obbligatori, ma sono una buona abitudine: fissando *una volta per tutte* le impostazioni più importanti del documento e avendo la precedenza sulle impostazioni dell'editor, evitano all'utente i problemi descritti nel paragrafo 3.2 a pagina 18.

Eccone la sintassi generale riconosciuta da TeXworks e TeXShop (ma non da Emacs):

```
%␣!TEX␣⟨parola chiave⟩␣=␣⟨valore⟩
```

dove:

- ⟨*parola chiave*⟩ è una tra le quattro voci spiegate più sotto;
- ⟨*valore*⟩ dipende dalla ⟨*parola chiave*⟩.

Si noti che maiuscole, minuscole e spazi, qui evidenziati con il segno ␣, vanno *sempre* rispettati (TeXShop permette di inserire queste righe anche automaticamente con appositi pulsanti).

Le possibili parole chiave sono:

`encoding` dichiara la codifica con cui è scritto il testo sorgente: si metta sempre `UTF-8` (verificando di aver caricato *anche* inputenc nello stesso modo, si veda il paragrafo 3.2 a pagina 18) e non si avranno problemi;

`program` dichiara il motore di composizione, che di solito è `pdflatex` o `xelatex` (tutto in minuscolo);

`root` dichiara il file principale e i file secondari nei documenti suddivisi in più file (si veda poco sotto);

`spellcheck` attiva il controllo ortografico della lingua del documento: per l'italiano si scriva `it-IT` (dopo aver installato i dizionari italiani).

TeXworks gestisce facilmente anche progetti complessi e in più file sorgente come una tesi o un libro (si veda il paragrafo 3.14 a pagina 41) permettendo di comporre comodamente il documento dalla finestra in cui ci si trova anziché tornare ogni volta al file principale. Immaginando di chiamare `Tesi.tex` il file principale della tesi su cui si sta lavorando, basta scrivere

```
% !TEX root = Tesi.tex
```

nella *prima riga* di *ciascun* file sorgente incluso nella tesi con `\input` se i file si trovano nella stessa cartella che contiene quello principale, oppure

```
% !TEX root = ../Tesi.tex
```

se i file si trovano in una sottocartella della cartella principale.

Si dà infine il blocco completo dei commenti speciali per un documento in italiano (l'ultimo non sempre serve):

```
% !TEX encoding = UTF-8
% !TEX program = pdflatex
% !TEX spellcheck = it-IT
% !TEX root = ⟨nome del file principale⟩
```

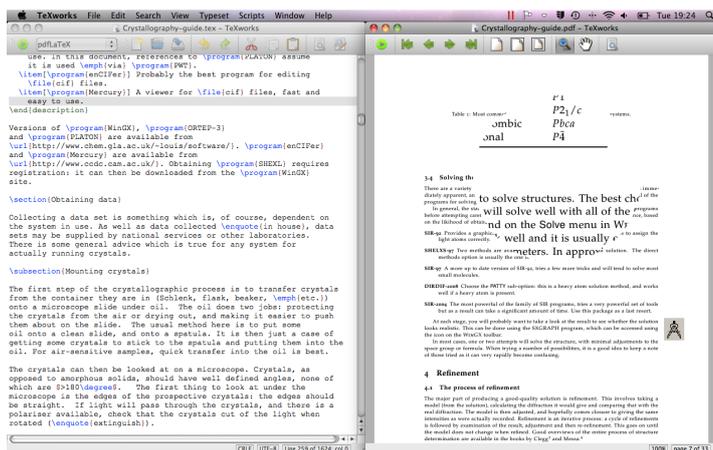



Figura 4: Una tipica schermata di TEXworks.

2.1.2 TEX Live

LATEX è uno, ma prende forma in differenti versioni (non così tante) che si chiamano *distribuzioni*. Una distribuzione è una raccolta di file e altro software autosufficiente per produrre un documento finito, comprendente il programma LATEX vero e proprio, uno o più editor dedicati e di solito altri programmi accessori. Si può installare direttamente da Internet, dal disco rigido dopo averla scaricata oppure da DVD.

Il luogo di riferimento nel Web da cui scaricare il materiale ufficiale su LATEX è CTAN (*Comprehensive TEX Archive Network*, "rete di archivi completi per TEX", ☞ CTAN), una rete di server dislocati in tutto il mondo, uguali tra loro e ciascuno contenente una copia integrale del sito originale (si chiamano anche *mirror*, "specchio"): ci si può così servire dal mirror più vicino a casa, evitando di sovraccaricare la rete e abbreviando così i tempi dell'operazione.

In questa guida si consiglia di installare TEX Live (☞ TEXLIVE): è una distribuzione completa, affidabile perché mantenuta da un gruppo di decine di sviluppatori e aggiornata annualmente (quella corrente è TEX Live 2011). Gli utenti di Windows possono contare sull'alternativa di MiKTEX (☞ MIKTEX), che gira solo su quel sistema operativo. Se la si sceglie, si abbia cura di installarla nella versione *Complete*.

TEX Live si modifica nel tempo, perché quasi ogni giorno molti dei *pacchetti* che la compongono (si veda il paragrafo 3.12 a pagina 39) vengono perfezionati, altri di nuovi vengono aggiunti agli archivi in Rete e altri ancora rimossi. Quelli già presenti riescono a soddisfare praticamente tutte le esigenze di scrittura, ma per mantenere la propria distribuzione sempre efficiente bisogna *aggiornarla* alle ultime versioni dei pacchetti e *installare* quelli nuovi continuamente creati, che perciò non vi figurano ancora.

Nei paragrafi seguenti si spiegherà:

- come installare TEX Live sui diversi sistemi operativi (si richiedono dai 3 ai 4 GiB di spazio sul disco rigido);

- come usare il programma `tlmgr` (*TeX Live ManaGeR*) per aggiornarla e installare automaticamente un nuovo pacchetto; si raccomanda di farlo senz'altro subito dopo aver installato la distribuzione e poi con una certa regolarità (una volta alla settimana potrebbe andare bene).

2.2 WINDOWS

2.2.1 Installare

La procedura descritta di seguito va eseguita con una connessione a Internet attiva e permette di installare sul proprio calcolatore TEX Live *aggiornata* alle ultime versioni dei pacchetti.

1. Si scarichi l'archivio compresso `install-tl.zip` da

   `http://www.tug.org/texlive/acquire-netinstall.html`

   e lo si decomprima.

2. Si apra la cartella risultante `install-tl-2011mmgg` (la data esatta è di solito quella corrente), si esegua *come amministratore* il file batch `install-tl` al suo interno e si attenda finché non compare l'omonima finestra Install-tl. (Si noti che Windows Vista e 7 richiedono *esplicitamente* i privilegi di amministratore: si clicchi sul file con il pulsante destro e si scelga Esegui come amministratore nel menù che compare.)

3. Si seguano le istruzioni sullo schermo senza cambiare nulla (si decida soltanto se installare o meno TEX Live per tutti gli utenti), si prema il pulsante Installa e si attenda la fine dell'operazione.

2.2.2 Aggiornare

L'interfaccia grafica di `tlmgr` su Windows è TEX Live Manager. Di seguito se ne dà la procedura per la prima volta. Per gli aggiornamenti successivi si salti il punto 2.

1. Si avvii il programma (Start → Programmi → TeX Live 2011 → TeX Live Manager).

2. Si apra la finestra Carica archivio pacchetti (tlmgr → Carica altro archivio...) e ci si assicuri che nella casella Archivio pacchetti: ci sia scritto

   `http://ctan.mirror.garr.it/mirrors/CTAN/systems/texlive/tlnet`

   In caso contrario, lo si scriva o lo si scelga premendo il pulsante Archivio dei pacchetti remoto predefinito e si dia Ok.

3. Si prema il pulsante Aggiorna installati in basso, si attenda la fine dell'aggiornamento e si chiuda il programma.

2.3 MAC OS X

2.3.1 Installare

Agli utenti di Mac OS X si raccomanda di installare TEX Live tramite MacTEX (☞ MACTEX), un installer che provvede a tutto il necessario. Segue la procedura.

1. Si scarichi l'archivio compresso `MacTeX.mpkg.zip` da

   `http://www.tug.org/mactex/2011/`

   e lo si decomprima.

2. Si esegua il pacchetto installer risultante `MacTeX-2011.mpkg` e se ne seguano le istruzioni fino a completare l'installazione.

2.3.2 Aggiornare

Su Mac OS X, TEX Live appena installata *non* è aggiornata alle ultime versioni dei pacchetti: va perciò aggiornata attraverso TEX Live Utility, l'interfaccia grafica di `tlmgr` per Mac. Di seguito si dà la procedura per la prima volta. Per gli aggiornamenti successivi si salti il punto 2.

1. Si avvii il programma (Applicazioni → TeX → TeX Live Utility).
2. Ci si assicuri che nella barra degli indirizzi in alto ci sia scritto

   `http://ctan.mirror.garr.it/mirrors/CTAN/systems/texlive/tlnet`

   In caso contrario, lo si scriva o lo si scelga dai mirror elencati alla voce Italy nella finestra Mirror Sites (Configure → Manage Mirros → EUROPE), si chiuda il programma e lo si riavvii.
3. Al termine della ricerca dei pacchetti, che il programma effettua automaticamente, si scelga la voce Update All Packages dal menù Actions, si attenda la fine dell'aggiornamento e si chiuda il programma.

2.4 LINUX

A causa di alcuni limiti imposti dagli sviluppatori non si può installare e aggiornare TEX Live sulle distribuzioni Linux con la stessa semplicità degli altri sistemi operativi. Le istruzioni descritte nella guida di [Gregorio, 2011] però, cui si rimanda senz'altro, permettono di farlo con estrema tranquillità. La procedura funziona su una distribuzione di Ubuntu correttamente installata e, con qualche modifica, anche su Fedora (e simili) e OpenSuSe (e simili).

A differenza di quanto accade su Windows e Mac OS X, su Linux TEX Live *non* comprende anche TEXworks, che va installato a parte tramite il gestore dei programmi.

2.5 ALBERO PERSONALE

Per usare un pacchetto che *non può* esserci nella distribuzione (quando è un pacchetto personale, o è coperto da una licenza particolare, o è una versione sperimentale che si vuole comunque provare, o ancora viene aggiornato o pubblicato durante il periodo di "congelamento") la strada più semplice è copiarne i file nella cartella di lavoro. (Ogni anno, infatti, qualche mese prima del rilascio della nuova versione la distribuzione viene "congelata" e gli aggiornamenti bloccati. Gli archivi in Rete, tuttavia, continuano a funzionare come al solito.)

Se si prevede di usarlo per più di un documento, invece, potrebbe essere più conveniente sistemarlo una volta per tutte nell'*albero personale*, una struttura ramificata di cartelle che parte dalla cartella `texmf`, la cui esatta ubicazione in ogni sistema operativo è mostrata nella tabella 3 nella pagina successiva.

Si noti che la cartella va creata, se non ci fosse già, e che fino al suo primo uso *rimane vuota*: l'utente dovrà ramificarla correttamente perché LATEX possa trovarvi senza problemi i file di cui ha bisogno. Di seguito si dà il percorso standard per poter usare un pacchetto esterno (è il caso più frequente):

`$HOME/texmf/tex/latex/`⟨*cartella*⟩

Tabella 3: Radice dell'albero personale nei diversi sistemi operativi.

OS	Versione	Radice
Windows	Vista	`C:\Documents and Settings\`⟨*nome utente*⟩`\Documenti\texmf\`
	XP	`C:\Documents and Settings\`⟨*nome utente*⟩`\texmf\`
	7	`C:\Users\`⟨*nome utente*⟩`\texmf\`
Mac OS X		`~/Library/texmf/`
Linux		`~/texmf/`

dove:

- `$HOME/texmf/` è la posizione dell'albero personale nel disco del proprio calcolatore;
- `/tex/latex/` sono le successive cartelle, da creare nell'ordine in cui appaiono nel percorso;
- ⟨*cartella*⟩ è l'ultima cartella eventualmente da creare, quella che contiene effettivamente i file del pacchetto in questione.

Per gli approfondimenti si rimanda a [Beccari, 2011].

3 BASI

Questo capitolo presenta alcune nozioni fondamentali indispensabili per comprendere il funzionamento di LaTeX. La sua lettura, perciò, è propedeutica al resto della guida.

3.1 PER COMINCIARE

In questo paragrafo si mostrano le semplici fasi per realizzare un documento con LaTeX, dalla scrittura del codice sorgente alla stampa.

Scrivere il codice

Innanzitutto si crei una cartella `prova` nella quale mettere *tutti* i file del documento. Dopo di che, con l'editor scelto si scriva il seguente testo (le istruzioni che contiene verranno spiegate nel corso di questo capitolo e in quello successivo):

```
\documentclass[a4paper]{article}
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}
\usepackage[italian]{babel}
\begin{document}
Ecco il mio primo documento con \LaTeX.
\end{document}
```

Si registri infine il file con il nome `primo.tex` (per convenzione, i file sorgente di LaTeX hanno l'estensione `.tex`).

Comporre

Si componga il codice premendo l'apposito pulsante. LaTeX mostra via via quello che sta facendo, emettendo contemporaneamente un messaggio di questo tipo (è il *log*, nel gergo di LaTeX), che in genere appare nella parte bassa dell'editor:

```
This is pdfTeX, Version 3.1415926 (TeX Live 2011)
(primo.tex
LaTeX2e <2009/09/24>
Babel <v3.8m> and hyphenation patterns for english, italian, loaded.
(/usr/local/texlive/2009/texmf-dist/tex/latex/base/article.cls
Document Class: article v1.4h Standard LaTeX document class
(/usr/local/texlive/2011/texmf-dist/tex/latex/base/fontenc.sty
(/usr/local/texlive/2011/texmf-dist/tex/latex/base/t1enc.def))
(/usr/local/texlive/2011/texmf-dist/tex/latex/base/inputenc.sty
(/usr/local/texlive/2011/texmf-dist/tex/latex/base/utf8.def))
(/usr/local/texlive/2011/texmf-dist/tex/generic/babel/babel.sty
(/usr/local/texlive/2011/texmf-dist/tex/generic/babel/italian.ldf
(/usr/local/texlive/2011/texmf-dist/tex/generic/babel/babel.def)))
[1] (primo.aux) )
Output written on primo.pdf (1 page, 15596 bytes).
Transcript written on primo.log.
```

Si noti che premere il pulsante di composizione dell'editor equivale a eseguire dalla riga di comando l'istruzione

```
pdflatex primo.tex
```

L'interfaccia a riga di comando (che su Windows si chiama *Prompt dei comandi*, su Mac *Terminale* e su Linux *Terminale* o *Console*) si avvia con modalità proprie di ogni sistema operativo.

In questo modo, si è consapevoli del fatto che è LaTeX (e *non* l'editor) il programma che elabora il codice `.tex`, che emette i messaggi e che produce il PDF, lavorando "dietro le quinte". La composizione non è dunque semplicemente "il tempo che bisogna attendere per vedere il PDF", ma il processo durante il quale il programma comprende le intenzioni dell'autore (espresse con i comandi), e le trasforma in un file (tipo)grafico.

Correggere gli errori

Se il codice sorgente contiene un errore (come il nome di un comando scritto scorrettamente), la composizione si arresta e nel log compare un messaggio più o meno di questa forma:

```
! Undefined control sequence.
l.6 Ecco il mio primo documento con \latex

?
```

Quando trova un errore, LaTeX indica:

- la natura dell'errore (in questo caso, `Undefined control sequence`, "sequenza di controllo non definita", ovvero "comando sconosciuto");
- la riga (*line*) del codice in cui si trova l'errore (`l.6`) ;
- il vero e proprio errore (il comando giusto è `\LaTeX`, non `\latex`: attenzione alle maiuscole!).

Per uscire dalla composizione basta dare `x` e `Invio`. Dopo aver corretto il codice, si registra il file e si ricompone (oppure, con alcuni editor, si ricompone direttamente).

Visualizzare e stampare

Se si usa un editor dedicato e il codice sorgente non contiene errori, la composizione andrà a buon fine e il visualizzatore di PDF si attiverà automaticamente mostrando il documento finito, che si potrà poi stampare con le modalità proprie del programma.

3.2 CODIFICHE, EDITOR E LINGUE

LaTeX è un programma nato per scrivere documenti *in inglese* ad alto contenuto matematico, e a questo scopo è stato originariamente corredato di una dotazione minima di caratteri (lettere, numeri e pochi altri segni) del tutto sufficiente, tant'è vero che il codice

```
\documentclass{⟨...⟩}
\begin{document}
...
\end{document}
```

permette di scrivere senza problemi un documento in quella lingua. La corretta scrittura di parole straniere, tuttavia, potrebbe richiedere di caricare i pacchetti descritti in questo paragrafo anche in un documento completamente in inglese.

Com'è noto, però, l'inglese si scrive senza accenti o caratteri particolari, contrariamente alla vasta maggioranza delle altre lingue che usano l'alfabeto latino. Gli unici caratteri "complicati" dell'italiano sono le vocali accentate, presenti (non proprio tutte) anche su una tastiera italiana. Perché LaTeX le riconosca come "valide" quando le si scrive con i tasti appositi, restituendole adeguatamente nel documento composto e sillabando correttamente le parole che le contengono, *immediatamente dopo* la dichiarazione di classe vanno caricati altri due pacchetti nell'ordine seguente:

```
\documentclass[⟨...⟩]{⟨...⟩}
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}
```

dove:

- T1 è la *codifica di output*, nel gergo di LaTeX, che serve a scrivere in italiano e in molte altre lingue occidentali. Per altre lingue o alfabeti particolari vanno usate altre codifiche: se un documento prevede parti in più lingue, si separeranno le diverse codifiche con una virgola, avendo cura di mettere *per ultima* quella della lingua principale: [T2A,T1] per un documento in italiano e russo, per esempio.

- fontenc (*font encoding*, "codifica dei font") si occupa dei font che si vedranno nel documento composto e fornisce a LaTeX i caratteri particolari di una certa lingua già disegnati.

- utf8 è la *codifica di input*, nel gergo di LaTeX, che permette di scrivere nell'editor direttamente dalla tastiera i segni di numerosi alfabeti evitando di dover caricare ogni volta la codifica adatta alla lingua del documento. Se servissero caratteri non compresi neppure in utf8 si può ampliare il campo con la codifica utf8x, da sola o in abbinamento al pacchetto ucs (si ricordi, in tal caso, di caricare ucs *prima* di inputenc). Se il proprio editor non supporta *pienamente* utf8, si usi latin1.

- inputenc (*input encoding*, "codifica di input") serve a LaTeX per interpretare correttamente il flusso dei caratteri che si immettono nell'editor.

Non ci sono molte alternative: senza questi due pacchetti le vocali accentate non verrebbero visualizzate nel documento composto. Grazie a inputenc, infine, il vecchio metodo che consisteva nel comporre *a mano* i caratteri con l'accento e che comportava non pochi problemi di sillabazione nelle parole che li contenevano, mostrato nell'esempio seguente,

```
Basta! Non se ne pu\'o pi\'u!
Perch\'e non c'\'e piet\'a
per chi deve scrivere cos\'i?
```

Basta! Non se ne può più! Perché non c'è pietà per chi deve scrivere così?

può essere finalmente abbandonato in favore di un codice più pulito:

```
Però: caricando il pacchetto
giusto la vita diventerà così
facile che c'è molto più gusto!
```

Però: caricando il pacchetto giusto la vita diventerà così facile che c'è molto più gusto!

Un ulteriore problema, spesso fonte di grattacapi e danni a volte irrimediabili, è il rapporto tra la codifica con cui è stato scritto il file sorgente e

quella con cui è impostato l'editor in uso, che *devono* coincidere, altrimenti aprendo un qualunque file sorgente ci si potrebbe trovare davanti a caratteri bizzarri. Perché ciò non accada, naturalmente, occorrerebbe conoscere in anticipo la codifica del file `.tex`, ma se non fosse possibile e si verificasse la situazione appena descritta, *non* si componga né si registri il file *per nessun motivo*, ma lo si chiuda, si reimposti correttamente l'editor e lo si riapra. Molti programmi di gestione dei testi oggi permettono di aprire file scritti in diverse codifiche e addirittura di decidere al momento di aprirli quale codifica si voglia usare: non dovrebbe essere un problema, se ne leggano con attenzione i manuali.

Si può rimediare ai danni appena prospettati sostituendo a mano i caratteri incomprensibili se il file è piccolo, oppure ricorrendo a un programma di conversione come Charco (☞ CHARCO) se invece è corposo.

Il terzo e ultimo pacchetto da caricare *sempre* è babel, che agisce su parole fisse (cioè le voci generate automaticamente dai comandi raccolti nella tabella 46), date, convenzioni tipografiche e sulla scelta delle regole di sillabazione, che per italiano e principali lingue europee sono già comprese in TeX Live. Come opzioni prende una o più lingue e per un documento in italiano lo si carica nel modo seguente:

```
\usepackage[⟨...⟩,italian]{babel}
```

dove l'*ultima* dichiarata è la lingua principale del documento. Inoltre, per ciascuna lingua definisce nuovi comandi per semplificare l'immissione dei caratteri particolari nazionali, come si può vedere nella documentazione del pacchetto o in [Gregorio, 2010].

Si dà, infine, il tipico inizio di un file sorgente per un documento in italiano con la corretta sequenza dei pacchetti da caricare:

```
\documentclass[⟨...⟩]{⟨...⟩}
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}
\usepackage[italian]{babel}
```

Il pacchetto babel definisce alcuni comandi molto utili per trattare correttamente ciascuna lingua in un documento multilingue. Supponendo di dover scrivere un documento in italiano con alcune parti in inglese, babel andrà caricato così:

```
\usepackage[english,italian]{babel}
```

Per singole parole o brevi frasi in lingua straniera è disponibile il comando

```
\foreignlanguage{⟨lingua⟩}{⟨testo⟩}
```

Per porzioni di testo in lingua più consistenti è disponibile l'ambiente

```
\begin{otherlanguage*}{⟨lingua⟩}
...
\end{otherlanguage*}
```

che però non traduce parole fisse e date. Se invece se ne avesse bisogno, si usi l'analogo ambiente `otherlanguage`.

3.3 FILE CON CUI SI HA A CHE FARE

Se si dà un'occhiata alla cartella di lavoro dopo la prima compilazione del testo sorgente, si noterà che oltre al file `.tex` iniziale e agli eventuali file esterni come le immagini ora sono comparsi altri file che prima non

Tabella 4: Principali file ausiliari di LaTeX.

Prodotti da	Estensione	Descrizione
Utente	`.bib`	Database bibliografico
	`.jpg`, `.pdf`, `.png`	Formati grafici per LaTeX
	`.tex`	File sorgente
Classi, pacchetti e stili	`.bst`	Stile bibliografico per BibTeX
	`.cls`	Classe di documento
	`.sty`	Pacchetto
Composizione	`.aux`	Trasporta informazioni generiche
	`.lof`	Indice delle figure
	`.log`	Rendiconta l'ultima composizione
	`.lot`	Indice delle tabelle
	`.toc`	Indice generale
Pacchetti e programmi	`.bbl`	Bibliografia creata con BibTeX
	`.blg`	Rendiconto di BibTeX
	`.idx`	Voci dell'indice analitico
	`.ind`	Prodotto di `MakeIndex`
	`.ilg`	Rendiconto di `MakeIndex`
	`.out`	Segnalibri ipertestuali
Output	`.pdf`	Prodotto di LaTeX

c'erano: sono i *file ausiliari* prodotti dalla composizione, e questo è un altro aspetto per cui LaTeX è molto diverso dai programmi tradizionali. Il loro numero varia in base alla complessità del documento, ma è importante sapere che a parte il file `.bib` *non* vanno toccati: LaTeX li crea e se ne serve automaticamente.

La tabella 4 raccoglie e descrive i principali file con cui si ha a che fare lavorando con LaTeX.

3.4 FILE SORGENTE

Un file sorgente di LaTeX è un file di puro testo che contiene sia il testo vero e proprio del documento sia i comandi che istruiscono LaTeX su come impostarne il formato.

3.4.1 Struttura del file sorgente

LaTeX si aspetta di trovare il file sorgente da elaborare strutturato in un certo modo. Elementi fondamentali sono almeno una dichiarazione di classe

```
\documentclass{⟨...⟩}
```

e le dichiarazioni d'inizio e fine documento:

```
\begin{document}
...
\end{document}
```

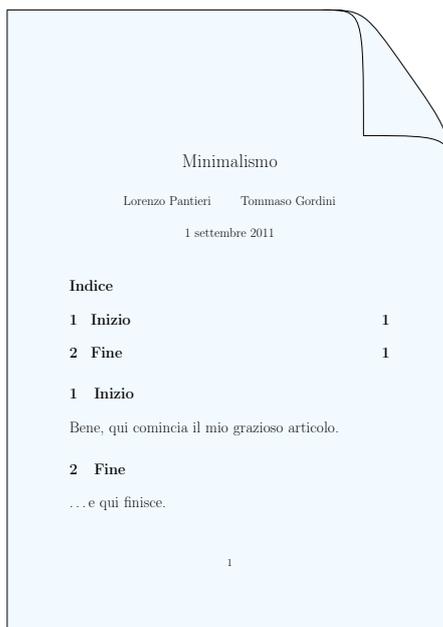
Tutte le istruzioni scritte tra `\documentclass` e `\begin{document}` *inclusi* costituiscono il *preambolo del documento* (o semplicemente *preambolo*) e comprendono:

- il caricamento di pacchetti che estendono le capacità di LaTeX;

Figura 5: Breve documento composto con LaTeX.

- le definizioni di comandi e ambienti personalizzati, che si consiglia di organizzare come indicato nel paragrafo 3.14 a pagina 41;

- le opzioni generali del documento.

Fra `\begin{document}` e `\end{document}` si scrive il testo vero e proprio, o *corpo del documento*, cioè tutto ciò che LaTeX elaborerà e si vedrà nel documento composto. Si noti che per isolare anche visivamente il corpo del documento dai due comandi appena descritti si può lasciare una riga bianca nel codice sorgente.

La figura 5 mostra un documento elementare composto da LaTeX a partire dal codice seguente:

```
\documentclass[a4paper]{article}
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}
\usepackage[italian]{babel}

\begin{document}

\author{Lorenzo Pantieri \and Tommaso Gordini}
\title{Minimalismo}
\maketitle

\tableofcontents

\section{Inizio}
Bene, qui comincia il mio grazioso articolo.

\section{Fine}
\dots e qui finisce.

\end{document}
```

Dove:

- `\begin{document}` segnala l'inizio del documento;
- `\author` e `\title` (che si possono dare anche prima del comando precedente) ne contengono rispettivamente il nome dell'autore e il titolo;
- `\and` si spiega da sé;
- `\maketitle`, che in ogni caso *deve seguire* i due comandi precedenti, ne stampa il contenuto;
- `\tableofcontents` ne stampa l'indice generale dopo *due* composizioni;
- `\section{`⟨*titolo*⟩`}` stampa un titolo di sezione (un paragrafo, in questo caso);
- `\dots` stampa i puntini di sospensione ... ;
- `\end{document}` segnala la fine del documento.

LaTeX ignora tutto ciò che trova dopo `\end{document}`: questo spazio, perciò, potrebbe essere un buon posto per appuntare un promemoria sul documento in elaborazione.

3.4.2 Spazi e righe vuote

Il modo in cui LaTeX tratta spazi, tabulazioni e righe vuote nel codice sorgente è particolarissimo, e decisamente diverso da quello di tutti i comuni elaboratori di testo. Infatti:

- una tabulazione è considerata come uno spazio;
- più spazi consecutivi sono considerati come un solo spazio;
- spazi o tabulazioni all'inizio di una riga vengono ignorati;
- una sola interruzione di riga è trattata come uno spazio;
- una riga vuota tra due righe di testo separa due capoversi;
- più righe vuote consecutive sono trattate come una sola riga vuota.

L'esempio seguente mostra il comportamento di LaTeX nei casi appena descritti; a sinistra c'è il codice sorgente, a destra il risultato della composizione:

```
Non ha alcuna importanza se si
mettono uno o tanti      spazi
dopo una parola.
       E neppure se si mettono
tanti spazi all'inizio di una
riga o se
la si interrompe.

Le cose cambiano se si saltano
una o più righe, perché in
questo modo si comincia un
nuovo capoverso.
```

Non ha alcuna importanza se si mettono uno o tanti spazi dopo una parola. E neppure se si mettono tanti spazi all'inizio di una riga o se la si interrompe.

Le cose cambiano se si saltano una o più righe, perché in questo modo si comincia un nuovo capoverso.

Tabella 5: Caratteri speciali di LaTeX.

Carattere	Funzione	Codice
\	Comincia un comando	\textbackslash
{ }	Delimitano un gruppo	\{ \}
$	Delimita la matematica in linea	\$
^	Esponente matematico	\^{}
_	Pedice matematico	_
&	Separa le celle in una tabella	\&
#	Numero dell'argomento	\#
~	Spazio indivisibile	\~{}
%	Commento	\%

3.4.3 Caratteri speciali

Le istruzioni di marcatura di LaTeX fanno largo uso di alcuni caratteri che il programma interpreta in modo particolare e che per la loro importanza si chiamano di solito *caratteri speciali*. Nella tabella 5 vengono mostrati insieme a funzione nel testo sorgente e codice per ottenerli in stampa. Nonostante la loro frequenza, però, possono dare qualche problema perché non hanno una posizione standard sulla tastiera oppure la loro scrittura richiede combinazioni di tasti o codici numerici particolari. Non tutti, inoltre, sono presenti su tutte le tastiere. Diventa tanto più importante, perciò, sapere come scriverli.

Si noti che:

- i caratteri { ' ~ } non sono presenti sulla tastiera italiana: la tabella 6 a fronte indica le scorciatoie da prendere in questi casi (su Windows, il codice relativo va digitato *sul tastierino numerico*);
- si distingua con attenzione ' (virgoletta alta aperta e accento grave) da ' (apostrofo, virgoletta alta chiusa e accento acuto);
- il comando \textbackslash *non* sostituisce la sequenza \\, come potrebbe sembrare: sono infatti due comandi distinti con distinte funzionalità (verranno considerati nei prossimi capitoli).

3.4.4 Comandi, dichiarazioni e ambienti

Un comune programma di videoscrittura a composizione sincrona e LaTeX presentano per certi versi un funzionamento simile: entrambi infatti ricevono dall'utente sia il testo sia i comandi per impostarne l'aspetto. La differenza, però, è sostanziale. Nel primo i comandi sono "preconfezionati": si scelgono da appositi menù e mostrano immediatamente i loro effetti, mentre le istruzioni che permettono di ottenerli rimangono nascoste. LaTeX, invece, con i suoi comandi le mantiene in superficie nei modi spiegati di seguito.

Un *comando* è un'istruzione che ordina a LaTeX di trattare in un certo modo una porzione più o meno ampia di testo. Si possono classificare i comandi di LaTeX in base alla forma che hanno e alla funzione che rivestono.

Per quanto riguarda la *forma* si distinguono tre tipi di comando, a seconda che siano formati da:

- Un solo carattere non alfabetico. Questi comandi sono quattro in tutto: spazio ␣, tilde ~, circonflesso ^ e trattino basso _.
- Una barra rovescia \ seguita da un solo carattere non alfabetico (cioè non compreso fra A-Z o a-z). Un comando di questo tipo termina al

Tabella 6: Scorciatoie da tastiera (italiana) per alcuni caratteri frequenti.

Carattere	Windows	Mac	Linux
`	Alt + 96	⌥ 9	Alt Gr + '
{	Alt + 123	⌥ ⇧ [	Alt Gr + 7
	Alt Gr + Maiusc + [		Alt Gr + Shift + [
}	Alt + 125	⌥ ⇧]	Alt Gr + 8
	Alt Gr + Maiusc +]		Alt Gr + Shift +]
~	Alt + 126	⌥ 5	Alt Gr + ì

primo carattere non alfabetico e uno o più spazi (che contano per un solo spazio) immediatamente dopo *non* vengono ignorati. Alcune tra le combinazioni più usate sono: `\{`, `\}`, `\%`, `\$`, `\_`, `\&`, `\#`, `\~`.

- Una barra rovescia \ seguita da una sequenza di caratteri alfabetici. Un comando di questo tipo termina al primo carattere non alfabetico e uno o più spazi immediatamente dopo vengono ignorati. Si noti che in questi comandi maiuscole e minuscole sono interpretate come *diverse*. Alcuni esempi: `\LaTeX`, `\emph`, `\documentclass`.

Si noti che per evitare una spaziatura errata dopo i comandi del terzo tipo che "producono testo" (come `\LaTeX`, `\TeX`, `\Ars`, `\dots`, `\today` e pochi altri) bisogna *necessariamente* farli seguire da qualcosa che *non* sia una lettera: un segno d'interpunzione, un comando, uno spazio esplicito, un gruppo vuoto `{}` o contenente altro testo. Altrimenti basta racchiuderli a loro volta in un gruppo, come mostra l'esempio seguente:

```
\Ars è la rivista del Gruppo
Utilizzatori Italiani di
\TeX\dots e \LaTeX,
naturalmente. \\
{\Ars} è la rivista del Gruppo
Utilizzatori Italiani di
\TeX\dots{} e \LaTeX,
naturalmente.
```

ArsTeXnicaè la rivista del Gruppo Utilizzatori Italiani di TeX. . . e LaTeX, naturalmente.
ArsTeXnica è la rivista del Gruppo Utilizzatori Italiani di TeX. . . e LaTeX, naturalmente.

La scrittura *sempre* corretta di questi comandi è una di quelle contenute nel secondo esempio. Si noti che un segno di punteggiatura immediatamente dopo il comando elimina la necessità dello spazio esplicito.

Per quanto riguarda la *funzione* si distinguono due tipi di comando, a seconda della porzione di testo su cui hanno effetto:

- Comandi come `\textit{`⟨*testo*⟩`}` ordinano a LaTeX di trattare in un certo modo solo il ⟨*testo*⟩ scritto tra le parentesi graffe.

- Comandi come `\itshape`, che si chiamano *dichiarazioni*, ordinano a LaTeX di trattare in un certo modo tutto *tutto* il testo successivo al punto in cui vengono dati.

In altre parole, una dichiarazione è un comando che imposta uno o più aspetti generali della composizione, e può essere data nel preambolo o nel corpo del documento. Nel primo caso il suo effetto si estende sull'intero documento e si annulla soltanto dando un'altra dichiarazione, nel secondo va data in un *gruppo* (cioè una porzione di testo racchiusa di solito da parentesi graffe o comandi di inizio e fine ambiente). Sono esempi di dichiarazione: `\small`, `\linespread`, `\appendix`. Nell'esempio seguente si vedono all'opera alcuni comandi visti fin qui:

Tabella 7: Principali tipi di comando in base a numero e tipo di argomenti richiesto. (F indica gli argomenti facoltativi, O quelli obbligatori.)

Argomenti		Esempio
F	O	
		`\LaTeX`
1		`\item[...]`
	1	`\emph{...}`
1	1	`\documentclass[...]{...}`
2	1	`\subfloat[...][...]{...}`
1	2	`\pdfbookmark[...]{...}{...}`
	3	`\multicolumn{...}{...}{...}`

```
Data odierna: \today. \\
Saro lì in \emph{dieci}
minuti. \\
Tutto il {\itshape testo
seguente è in corsivo}.
```

Data odierna: 19 febbraio 2012.
Saro lì in *dieci* minuti.
Tutto il *testo seguente è in corsivo*.

Si noti che il comando `\today` produce la data in cui si compone il documento secondo le tradizioni della lingua in uso.

La giustapposizione degli elementi di un comando prende il nome di *sintassi* del comando. Ciò che va tra parentesi graffe si chiama *argomento obbligatorio*, mentre ciò che va tra parentesi quadre si chiama *argomento facoltativo*. Se gli elementi da scrivere all'interno dello stesso gruppo di parentesi sono più d'uno, vanno separati con una virgola *senza ulteriori spazi*. La tabella 7 mostra i principali tipi di comando in base a numero e tipo di argomenti che richiedono.

Un *ambiente* è una porzione di codice delimitata da un comando d'apertura e uno di chiusura che LaTeX tratta in un certo modo. La sintassi di un ambiente generico è:

```
\begin{⟨ambiente⟩}[⟨...⟩]{⟨...⟩}
...
\end{⟨ambiente⟩}
```

dove:

- ⟨*ambiente*⟩ è il nome dell'ambiente;
- se presenti, argomenti facoltativi e obbligatori si scrivono dopo il solo comando d'apertura;
- l'ambiente va separato dal resto del testo con una riga bianca prima e dopo se ciò che contiene *non* appartiene al flusso del discorso (una figura, per esempio); non va separato in caso contrario.

3.4.5 Commenti

Quando LaTeX incontra un carattere di percento `%` (tranne che nella forma `\%`) elaborando un file sorgente, ignora il resto della riga, l'interruzione di riga, e tutti gli spazi bianchi all'inizio della riga successiva. Questo carattere può dunque essere impiegato per appendere un promemoria che non verrà stampato.

Talvolta però va usato per spezzare parole troppo lunghe o per dividere righe in cui non sono permessi spazi bianchi o interruzioni (come nel caso di comandi troppo lunghi per stare in una sola riga dell'editor). Eccolo all'opera:

```
Ecco un % semplice,
% ma istruttivo
esempio: Supercal%
              ifragilist%
    ichespiralidoso.
```

Ecco un esempio: Supercalifragilistichespiralidoso.

Per commenti più lunghi si può usare l'ambiente `comment` definito dall'omonimo pacchetto:

```
Ecco un altro
\begin{comment}
semplice,
ma utile
\end{comment}
esempio per includere commenti
nel proprio documento.
```

Ecco un altro esempio per includere commenti nel proprio documento.

3.4.6 Codici sorgente ordinati

Spesso gli utenti di LaTeX sottovalutano l'importanza di un codice sorgente "pulito" (con rientri, incolonnamenti, eccetera) e commentato. *Non* è indispensabile, ma si consiglia di farlo ugualmente: l'ordine facilita la gestione del codice, soprattutto se a uno stesso progetto lavorano più persone, e rende più facile ricercare eventuali errori.

Durante la stesura si raccomanda di usare i commenti, di suddividere con chiarezza il documento e di aiutarsi eventualmente con rientri, incolonnamenti, a capo e righe vuote supplementari. Ulteriori consigli in questo senso verranno forniti nelle prossime pagine al momento opportuno.

Si esemplificano ora le indicazioni precedenti in un codice sorgente "ben scritto".

Esempio di articolo composto con LaTeX.

```
% Un articolo scritto con LaTeX
\documentclass[a4paper,11pt]{article}
\usepackage[T1]{fontenc}      % imposta la codifica dei font
\usepackage[utf8]{inputenc}   % lettere accentate da tastiera
\usepackage[italian]{babel}   % per scrivere in italiano
\usepackage{layaureo}         % imposta i margini di pagina
\usepackage{lipsum}           % genera testo fittizio
\usepackage{url}              % per scrivere gli indirizzi Internet

\begin{document}

\author{Lorenzo Pantieri \and Tommaso Gordini}
\title{Il titolo}
\maketitle

\begin{abstract}
\lipsum[1]
\end{abstract}

\tableofcontents

\section{Un paragrafo}
\lipsum[1]

\subsection{Un sottoparagrafo}
\lipsum[1]
```

```
\section{Un paragrafo}
\label{sec:esempio}
\lipsum[1]

% Bibliografia
\begin{thebibliography}{9}
\bibitem{pantieri:arte}
Pantieri, Lorenzo e Tommaso Gordini (2011),
\emph{L’arte di scrivere con \LaTeX},
\url{http://www.lorenzopantieri.net/LaTeX_files/ArteLaTeX.pdf}.
\end{thebibliography}

\end{document}
```

3.5 CLASSI DI DOCUMENTO

La prima informazione che LaTeX richiede quando elabora un file sorgente è il *tipo di documento* che si desidera realizzare (la *classe*, nel gergo di LaTeX), da specificare *come prima riga del codice* con il comando

```
\documentclass[⟨opzioni⟩]{⟨classe⟩}
```

dove:

- ⟨*opzioni*⟩ sono le impostazioni generali del documento (da separare con la virgola, se più d'una);
- ⟨*classe*⟩ è la classe di documento con cui si intende scrivere il proprio lavoro.

Di seguito si elencano le principali *classi standard* di documento (cioè già definite dal programma):

article per scrivere articoli;

report per scrivere relazioni o tesi strutturate in diversi capitoli e dotate eventualmente di un sommario;

book per scrivere libri;

letter per scrivere lettere.

Esistono numerose altre classi aggiuntive per i più diversi documenti, compresi lucidi e presentazioni. Tra le classi non standard più diffuse ci sono memoir (che consente di personalizzare qualunque aspetto del documento in modo particolarmente versatile), toptesi e suftesi (specifiche per tesi di laurea e dottorato), beamer (per le presentazioni). Per esempio, questa guida è scritta con lo stile ClassicThesis basato sulle classi KOMA-Script, leggermente diverse da quelle standard (per maggiori dettagli si veda [Pantieri, 2007]).

Le ⟨*opzioni*⟩ date a `\documentclass` si dicono anche *globali* e regolano il comportamento generale della classe di documento. Di seguito si descrivono quelle più comuni per le tre classi article, report e book (la classe letter meriterebbe una trattazione a parte, perciò non viene considerata). Si noti che alcune di esse hanno un'applicabilità limitata, come mostra la tabella 8 a fronte.

10pt, **11pt**, **12pt** Impostano la dimensione del font principale del documento. Omettendo l'opzione, il valore predefinito è `10pt`.

Tabella 8: Opzioni più comuni delle classi standard di LaTeX. I simboli ●, ◐ e ○ indicano rispettivamente che l'opzione è predefinita, applicabile (anche se non predefinita), non applicabile.

Opzione	book	report	article
10pt	●	●	●
letterpaper	●	●	●
oneside	◐	●	●
twoside	●	◐	◐
openany	◐	●	○
openright	●	◐	○
titlepage	●	●	◐
final	●	●	●

a4paper, **a5paper**, ... Definiscono le dimensioni del foglio, che per impostazione predefinita è nel formato americano `letterpaper`. Le altre opzioni possibili sono `executivepaper`, `legalpaper` e `b5paper`.

oneside, **twoside** Specificano se verrà composto un documento a singola o doppia facciata rispettivamente. Per impostazione predefinita, le classi article e report sono a singola facciata e la classe book è a doppia facciata.

openany, **openright** L'opzione `openany`, predefinita nella classe report, fa iniziare i capitoli nella successiva pagina a disposizione; `openright`, predefinita nella classe book, fa iniziare i capitoli sempre in una pagina destra. Entrambe non sono disponibili nella classe article, che non ammette la suddivisione in capitoli.

twocolumn Dà a LaTeX le istruzioni per comporre l'intero documento su due colonne (si veda anche il paragrafo 9.3 a pagina 133).

titlepage, **notitlepage** Specificano se dopo il titolo del documento debba avere inizio una nuova pagina (come accade con report e book) o meno (come accade con article) rispettivamente.

fleqn Allinea le formule a sinistra rispetto a un margine rientrato.

leqno Mette la numerazione delle formule a sinistra anziché a destra.

draft, **final** L'opzione `draft` evidenzia le righe composte in modo non ottimale con un rettangolino nero ▮ accanto, rendendole più facilmente individuabili. Ciò non accade con `final`. Si noti che entrambe influenzano il comportamento degli altri pacchetti caricati o addirittura li disabilitano del tutto (si veda il paragrafo 10.1 a pagina 143 per risolvere il problema).

Il codice sorgente di un documento da comporre con LaTeX, allora, potrebbe iniziare con la riga

```
\documentclass[a4paper,11pt,twoside]{article}
```

che ordina a LaTeX di impaginare il documento come un articolo, su carta di formato A4, con un carattere di 11 punti e impostato per la stampa fronte/retro.

3.6 GESTIRE LA PAGINA

3.6.1 Il tormentone dei margini

I margini di pagina in tipografia rivestono funzioni importantissime, tra cui quella di delimitare in modo chiaro il corpo del testo per permettere al lettore di individuarlo senza difficoltà e maneggiarlo comodamente appoggiando i pollici su uno spazio sufficientemente confortevole [Bringhurst, 1992]. Questa è la ragione principale per cui i margini esterni predefiniti nei documenti impostati per la stampa in fronte/retro sono più ampi di quelli interni, che appaiono duplicati perché adiacenti.

La maggior parte degli utenti europei, che stampa su carta in formato A4, ritiene troppo ampi i margini predefiniti da LaTeX nelle classi standard, e che di conseguenza la pagina non sia sufficientemente riempita. Prima di buttarsi nella frenesia dell' "allarghiamo un po' questa strettissima pagina", però, è doveroso riflettere.

I margini di LaTeX derivano da convenzioni tipografiche ampiamente verificate e accettate, e mettono l'autore nelle vantaggiose condizioni di potersene servire per ottenere risultati professionali *già alla prima composizione* e senza doverci mettere le mani. Modificarli, perciò, significherebbe dover studiare un (bel) po' di tipografia prima di raggiungere risultati accettabili.

L'esperienza dimostra che leggere diventa tanto più difficile quanti più caratteri ci sono in una singola riga di testo: l'occhio è costretto a compiere movimenti più ampi, e si affatica presto (questo è il motivo per cui quotidiani e riviste sono stampati su più colonne). Robert Bringhurst ha codificato quest'esperienza nella sua celebre "regola", che considera ottimale il numero di circa 66 caratteri per riga (spazi inclusi), indipendentemente dal font usato [Pantieri, 2007]. L'ampiezza media in punti di un carattere corrisponde al rapporto tra la lunghezza dell'alfabeto latino minuscolo "abcdefghijklmnopqrstuvwxyz" e il numero di lettere che lo compongono (26). Se si considera che per riempire meglio la pagina LaTeX usa già *in partenza* una riga più lunga del limite stabilito da Bringhurst, diventa evidente come si debbano evitare il più possibile modifiche di questo tipo.

In alcune circostanze, tuttavia, può essere desiderabile o necessaria una copertura della pagina ancora maggiore: tra i numerosi pacchetti scritti a questo scopo se ne consigliano due.

Il pacchetto LayAureo (se ne veda la documentazione in italiano) definisce un layout di pagina pronto per l'uso, permettendo di impostare facilmente anche lo spazio per la rilegatura con la chiave `binding=`⟨*dimensione*⟩. Il pacchetto agisce semplicemente caricandolo e non è personalizzabile: dunque, o piace o non piace.

Se fossero necessarie proporzioni di pagina ancora diverse (perché la propria facoltà impone un modello di tesi particolare, per esempio) da applicare anche per una sola pagina, il pacchetto geometry, completamente configurabile, può risolvere il problema.

Si immagini di dover comporre un documento in formato A4 con margini superiore e inferiore di 3 cm, sinistro e destro (che nella stampa in fronte/retro diventano interno ed esterno) di 3,5 cm e di voler destinare alla rilegatura uno spazio di 5 mm. Il codice da scrivere nel preambolo è il seguente:

```
\usepackage{geometry}
\geometry{a4paper,top=3cm,bottom=3cm,left=3.5cm,right=3.5cm,%
          heightrounded,bindingoffset=5mm}
```

Tra le opzioni del pacchetto che, si noti bene, come il precedente agisce anche solo caricandolo, si consiglia sempre anche `heightrounded`, che mo-

difica ulteriormente di poco le dimensioni della gabbia del testo per farle contenere un numero intero di righe.

Si eviti *nel modo più assoluto*, invece, di toccare comandi interni di LaTeX come `\textwidth`, `\oddsidemargin`, eccetera, perché la loro azione non tiene in nessun conto le proporzioni di pagina [Fairbairns, 2011].

Talvolta il rilegatore potrebbe chiedere dove deve tagliare il foglio: il pacchetto crop permette di ottenere sul documento i crocini di taglio.

3.6.2 Interlinea e riempimento della pagina

Interlinea

L'interlinea standard di LaTeX (fissata a 1) garantisce un risultato tipografico ottimale e non andrebbe modificata senza una ragione precisa. Purtroppo, però, esistono editori e regolamenti di facoltà che impongono di scrivere pubblicazioni e tesi in modo non professionale. Per modificarla si consiglia il pacchetto setspace, che definisce tre interlinee *globali* da impostare *nel preambolo* come segue:

- `\singlespacing` (interlinea 1);
- `\onehalfspacing` (interlinea 1,5);
- `\doublespacing` (interlinea 2).

Si può modificare l'interlinea soltanto in alcune parti del documento con gli ambienti `singlespace`, `onehalfspace` e `doublespace`, da usare nel modo consueto.

Se, infine, ne fosse necessaria una ancora diversa, si può risolvere il problema dando *nel preambolo* il comando standard

```
\linespread{⟨fattore di scala⟩}
```

che moltiplica l'interlinea per il ⟨*fattore di scala*⟩. In alternativa, il pacchetto definisce l'ambiente `spacing` da impostare come `\linespread`:

```
\begin{spacing}{⟨fattore di scala⟩}
...
\end{spacing}
```

Riempimento della pagina

In mancanza di istruzioni specifiche LaTeX cerca sempre di riempire la gabbia del testo per tutta la sua altezza. Se non riesce a farlo perché non c'è abbastanza testo, "stiracchia" il materiale che ha a disposizione inserendo dello spazio aggiuntivo dove può: tra i capoversi, tra le voci degli elenchi e così via. Si può disattivare questo comportamento per ottenere dello spazio bianco in fondo alla pagina (scelta sconsigliata) scrivendo nel preambolo `\raggedbottom`.

3.7 STRUTTURARE IL DOCUMENTO

La tabella 9 nella pagina successiva raccoglie le principali istruzioni che producono una sezione nel documento e ne descrive il comportamento nelle classi standard. Non si considera la classe letter poiché non prevede alcun tipo di sezionamento.

Tabella 9: Istruzioni che producono una sezione nel documento e loro comportamento nelle classi standard. I simboli indicano che la sezione possiede la caratteristica sempre (●), mai (○), solo nella classe article (◐), solo nelle classi article e report (◑). Tra parentesi i pacchetti richiesti.

		La sezione prodotta ha				
A cosa serve	Istruzione	numero	titolo	testatine	posto nell'indice	Sezione
Corpo del testo	`\part`	●	●	○	●	Parte
	`\chapter`	●	●	●	●	Capitolo
	`\section`	●	●	●	●	Paragrafo
	`\subsection`	●	●	○	●	Sottoparagrafo
	`\subsubsection`	◐	●	○	◐	Sotto-sottoparagrafo
	`\paragraph`	○	●	○	○	Sezione di livello ancora più basso
	`\subparagraph`	○	●	○	○	Sezione al più basso livello possibile
Indici	`\tableofcontents`	○	●	●	○	Indice generale
	`\listoffigures`	○	●	●	○	Elenco delle figure
	`\listoftables`	○	●	●	○	Elenco delle tabelle
	`\printindex`	○	●	●	○	Indice analitico (richiede makeidx)
Bibliografia	`thebibliography`	○	●	●	○	Bibliografia manuale
	`\printbibliography`	○	●	●	○	Bibliografia automatica (richiede biblatex)
Varie	`abstract`	○	◑	○	○	Sommario

3.7.1 Sezionare il corpo del testo

Per suddividere un documento in capitoli, paragrafi e sezioni di livello più basso basta dare *nel corpo del testo* i comandi elencati nella prima parte della tabella, a proposito dei quali si noti quanto segue.

- I nomi inglesi dei comandi corrispondono alle unità di suddivisione del testo in vigore nei Paesi anglosassoni, e alcuni di essi non hanno corrispondenti italiani.
- Il comando `\section` produce una sezione equivalente al nostro *paragrafo*, e `\paragraph` ("capoverso", in inglese) una sezione non numerata non equivalente né al nostro *paragrafo* né al nostro *capoverso* (che ammette al massimo un titoletto o una breve indicazione). Le stesse considerazioni valgono anche per i rimanenti comandi.
- Il comando `\part` non influenza la numerazione progressiva dei capitoli.
- Si consiglia di evitare suddivisioni così fini come quelle permesse dagli ultimi tre comandi e di assicurarsi che ogni sezione contenga *almeno due* sezioni di livello immediatamente inferiore: in caso contrario, il comando di sezionamento diventa superfluo.

3.7.2 Altri sezionamenti

Le altre istruzioni mostrate nella tabella 9 a fronte producono le corrispondenti sezioni descritte nei prossimi capitoli. Si noti che l'ambiente abstract, destinato a ospitare il sommario (o riassunto) di un lavoro, è ammesso solo nelle classi article e report perché di solito nei libri il sommario è sostituito dall'introduzione.

3.7.3 Materiale iniziale, principale e finale

Oltre ai comandi appena illustrati, la sola classe book prevede tre dichiarazioni che agiscono al più alto livello possibile e costituiscono una specie di "super-sezionamento". Vanno date *sempre* nel corpo del documento e si comportano come segue:

- `\frontmatter` ("materiale iniziale") non numera le sezioni e numera le pagine con numeri romani minuscoli (i, ii, iii, eccetera);
- `\mainmatter` ("materiale principale") numera le sezioni e le pagine con numeri arabi (la numerazione della pagine riprende da 1);
- `\backmatter` ("materiale finale") non numera le sezioni e continua la numerazione araba delle pagine dal materiale principale.

3.7.4 Appendici

Per produrre le appendici basta dare la dichiarazione `\appendix`, che cambia i numeri dei capitoli (o dei paragrafi, se la classe impostata è article) in lettere.

Il pacchetto appendix permette eventualmente di personalizzarle (se ne veda la documentazione).

Tabella 10: Struttura generale di un libro o una tesi. Le voci in corsivo sono obbligatorie, quelle in tondo sono facoltative, quelle asteriscate *non* devono comparire nell'indice generale.

Supersezionamento	Sezione
Materiale iniziale	*Frontespizio* Colophon* Dedica* Sommario* *Indice generale** Elenco delle figure* Elenco delle tabelle* Altri elenchi* Prefazione Ringraziamenti* Introduzione non numerata
Materiale principale	Introduzione numerata *Capitoli* Una o più appendici numerate
Materiale finale	Una o più appendici non numerate Glossario *Bibliografia* Indice analitico

3.7.5 Struttura generale di un libro o una tesi

La tabella 10 mostra una *possibile* successione dei componenti di una pubblicazione di una certa consistenza, come un libro o una tesi di laurea o dottorato. Lo schema è tratto con qualche variazione da [Mori, 2007], cui si rimanda per ogni approfondimento.

Si noti che:

- l'introduzione va senz'altro nel materiale principale se è a propria volta divisa in sezioni, mentre va in quello iniziale se è breve e contiene solo una sintetica esposizione dell'argomento;
- le appendici vanno valutate caso per caso in base a numero e importanza: se è una sola e poco importante può andare nel materiale finale; se invece è funzionale al corpo principale del documento va in quello principale.

Si ricordi, infine, che una sezione che si concepisce come un capitolo non va *mai* messa nel materiale finale.

3.8 STILI DI PAGINA

Lo *stile di pagina* è l'organizzazione del contenuto di testatina e piede scelta per il documento, e va indicato nell'argomento ⟨*stile*⟩ del comando

```
\pagestyle{⟨stile⟩}
```

LaTeX prevede tre stili di pagina predefiniti e uno personalizzabile, descritti di seguito.

`plain` Stampa i numeri di pagina nel piede, lasciando vuota la testatina. È lo stile predefinito nelle classi article e report.

empty Lascia testatina e piede vuoti.

headings Lascia il piede vuoto e compone le testatine come segue: il numero di pagina è sempre posto nel margine esterno, seguito dal titolo del capitolo corrente nella testatina di sinistra e preceduto dal titolo del paragrafo corrente in quella di destra. È lo stile predefinito nella classe book e agisce nello stesso modo nella classe report e article impostate con `twoside`, con la differenza che in quest'ultima le testatine riportano i titoli del paragrafo e del sottoparagrafo correnti, rispettivamente. Se invece si imposta la classe con `oneside`, la testatina riporta soltanto il titolo della suddivisione maggiore.

myheadings È simile a `headings` nel risultato e va usato quando non si vuole che le testatine dipendano dai titoli delle sezioni (capitolo e paragrafo) correnti. L'autore deve specificarne il contenuto a ogni nuovo capitolo (o paragrafo, se la classe è article), dando `\markboth` per comporle entrambe oppure `\markright` per comporre soltanto quella di destra.

Si può cambiare lo stile della pagina *corrente* con il comando

```
\thispagestyle{⟨stile⟩}
```

Il pacchetto fancyhdr (se ne veda la documentazione), infine, permette di regolare finemente ogni aspetto dello stile di pagina.

Gestire le testatine

Nelle classi standard le testatine vengono prodotte dai comandi mostrati nella tabella 9 a pagina 32. La loro presenza o meno sulla pagina si può regolare con facilità, ma si raccomanda di non abusare di queste possibilità e di attenersi alla scelta tipografica della classe in uso. Si ricorda che le dichiarazioni di sezionamento descritte nel paragrafo 3.7.3 a pagina 33 *non* influenzano il comportamento delle testatine, che saranno dunque sempre presenti anche nelle sezioni non numerate prodotte da `\frontmatter`.

Per eliminare le testatine da una sezione non numerata *dall'autore*, invece, basta impostare per quella sezione lo stile di pagina `plain` (si veda il paragrafo 3.8 a fronte) e ripristinare poi lo stile generale come segue:

```
\chapter*{Prefazione}
\pagestyle{plain}
...

\chapter{Introduzione}
\pagestyle{headings}
...
```

Si noti bene che se ci si dimentica di farlo, la sezione non numerata porterà le testatine prodotte dall'ultimo comando "utile" in questo senso.

Se comunque le si volessero, vanno inserite a mano con `\markboth`

```
\markboth{\MakeUppercase{⟨testatina di sinistra⟩}}%
         {\MakeUppercase{⟨testatina di destra⟩}}
```

che si usa come segue:

```
\chapter*{Prefazione}
\markboth{\MakeUppercase{Prefazione}}{\MakeUppercase{Prefazione}}
```

dove `\MakeUppercase` produce il proprio argomento tutto in maiuscolo.

Eliminare testatine inutili

Per eliminare le testatine e i piedi che *comunque* si avranno nelle pagine bianche alla fine di un capitolo usando le classi standard impostate con l'opzione `openright` (scelta consigliata), basta caricare il pacchetto emptypage nel modo consueto.

3.9 INDICI E TITOLI ALTERNATIVI

3.9.1 Indice generale

Il comando

```
\tableofcontents
```

produce *nel punto in cui viene dato* la sezione contenente l'indice generale con la relativa testatina. Si noti che per ottenerlo nel documento finito servono *due* composizioni successive.

In particolari circostanze potrebbe essere utile inserire anche un *miniindice* fra titolo del capitolo e inizio del testo: il pacchetto minitoc permette di farlo automaticamente (se ne veda la documentazione), ma non se ne abusi.

3.9.2 Gestire i titoli

Titoli non numerati né indicizzati

Di tutti i comandi di sezionamento elencati nella tabella 9 a pagina 32 esiste anche una *variante asterisco* formata dal comando e da un asterisco alla fine, che genera titoli *non numerati* e che nemmeno andranno a finire nell'indice generale, di qualunque livello essi siano. Il titolo precedente si è ottenuto con il comando

```
\subsubsection*{Titoli non numerati né indicizzati}
```

Per mandare nell'indice anche i titoli che normalmente non ci finirebbero, *subito dopo* il relativo comando di sezionamento basta dare

```
\addcontentsline{toc}{⟨livello⟩}{⟨titolo⟩}
```

dove:

- `toc` è il tipo di indice in cui far comparire la voce in questione (l'indice generale, in questo caso);
- ⟨*livello*⟩ è il nome del livello di sezionamento in questione (si noti che `paragraph` e `subparagraph` non sono ammessi);
- ⟨*titolo*⟩ è il titolo di sezione che finirà nell'indice.

Applicato al titolo di questa sezione, il codice sarebbe:

```
\subsection*{Titoli non numerati né indicizzati}
%\phantomsection
\addcontentsline{toc}{subsection}{Titoli non numerati né indicizzati}
```

Si noti che se hyperref è caricato, *subito prima* di `\addcontentsline` bisogna dare anche `\phantomsection` per evitare possibili errori nei collegamenti ipertestuali e nei segnalibri del documento finito (in tal caso, si decommenti la riga corrispondente).

Titoli alternativi nell'indice generale

Nell'indice generale finiscono i titoli scritti nell'argomento dei comandi di sezionamento. Se un titolo, però, è troppo lungo per starci agevolmente (si noti, però che un titolo non dovrebbe *mai* andare a capo) o si hanno particolari esigenze, lo si può sostituire con un titolo alternativo più breve, da inserire nell'argomento facoltativo degli stessi comandi:

```
\chapter[Leggilo! È emozionante!]{Questo è un titolo lunghissimo
         e particolarmente noioso}
```

Si noti che il titolo breve comparirà anche nelle testatine, se previste dalla classe di documento in uso e che, ovviamente, non si può usare se il comando è asteriscato.

3.10 RIFERIMENTI INCROCIATI

Nei documenti si trovano spesso riferimenti incrociati a sezioni, figure, tabelle, teoremi e altri elementi. Per realizzarli si usano i comandi standard

```
\label{⟨etichetta⟩}
\ref{⟨etichetta⟩}
\pageref{⟨etichetta⟩}
```

dove:

- ⟨*etichetta*⟩ è un identificatore *univoco* scelto dall'autore il cui formato è libero, ma si consiglia di adottare la forma ⟨*abbreviazione*⟩:⟨*parola chiave*⟩ (scrivendo in francese si sconsiglia di mettere i due punti nelle etichette);
- `\ref` stampa il numero della sezione o dell'oggetto messo in ⟨*etichetta*⟩;
- `\pageref` si comporta come `\ref`, stampando però solo il numero di pagina in cui l'oggetto compare.

Proprio come accade con l'indice, per attivare i riferimenti incrociati nel documento sono necessarie *due* composizioni successive.

Per esempio, se si identifica questo paragrafo con

```
\section{Riferimenti incrociati}
\label{sec:rif-inc}
```

poi ci si può riferire a esso con

```
Ecco un riferimento a questo
paragrafo: ''si veda il
paragrafo~\ref{sec:rif-inc}''.
```

Ecco un riferimento a questo paragrafo: "si veda il paragrafo 3.10".

Si noti che è una buona abitudine unire il riferimento alla parola precedente con uno *spazio indivisibile*: garantisce in genere risultati tipografici ottimali.

Molto spesso, specialmente quando il riferimento si trova a una o più pagine di distanza dall'oggetto, è utile avere un'indicazione completa *anche* del numero di pagina. Lo si può ottenere con il comando `\vref` del pacchetto varioref dopo aver messo la lingua principale del documento come opzione al comando di chiamata:

```
\usepackage[italian]{varioref}
```

A questo punto `\vref` si usa come di consueto:

```
Si veda il
paragrafo~\vref{sec:rif-inc}.
```

Si veda il paragrafo 3.10.

3.11 COLLEGAMENTI IPERTESTUALI E AL WEB

Collegamenti ipertestuali e al Web

Il pacchetto hyperref, che di regola va caricato *per ultimo*, crea i collegamenti ipertestuali all'interno del documento, rendendo cliccabili i riferimenti incrociati visti nella sezione precedente, quelli a voci bibliografiche, a indirizzi Internet e molto altro. Se ne possono indicare le opzioni come di consueto oppure, se numerose, si può usare il comando `\hypersetup`:

```
\usepackage{hyperref}
\hypersetup{⟨chiave⟩=⟨valore⟩,⟨...⟩}
```

Il pacchetto prevede numerose opzioni per gestire collegamenti e segnalibri in modo molto fine: se ne veda la documentazione.

Per impostazione predefinita, hyperref circonda il collegamento con un riquadro colorato che *non* viene stampato. Si può avere il testo del collegamento colorato (con i colori predefiniti o a piacere) scrivendo:

```
\usepackage[colorlinks]{hyperref}
```

Questo è utile per un documento da leggere a schermo o da stampare a colori, ma si ricordi che la stampa in bianco e nero restituisce i colori come sfumature di grigio, a volte poco leggibili.

Si possono avere tutti i collegamenti in nero e senza riquadri scrivendo semplicemente

```
\hypersetup{hidelinks}
```

Per una gestione più fine dei segnalibri, che con l'opzione `bookmarks` hyperref gestisce automaticamente, compresi quelli delle sezioni aggiunte all'indice generale con `\addcontentsline`, si segnala il pacchetto bookmarks, da caricare *dopo* hyperref (se ne veda la documentazione).

Oltre ai collegamenti ipertestuali per i riferimenti incrociati, hyperref permette anche di realizzare collegamenti al Web con il comando `\href`:

```
\href{⟨indirizzo Internet⟩}{⟨testo del collegamento⟩}
```

Scrivendo

```
Visita il sito del
\href{http://www.guitex.org/}
    {\GuIT*}.
```

Visita il sito del GuIT.

basta cliccare sul logo GuIT per accedere al sito omonimo.

Indirizzi Internet e di posta elettronica

Il pacchetto url (caricato automaticamente da hyperref) definisce il comando `\url`, utile per scrivere un indirizzo Internet:

```
\url{http://profs.sci.univr.%
    it/~gregorio/}
```

http://profs.sci.univr.it/~gregorio/

Per i collegamenti a un indirizzo di posta elettronica conviene definire nel preambolo un apposito comando `\mail` (si veda il paragrafo 9.1 a pagina 127),

```
\newcommand{\mail}[1]{\href{mailto:#1}{\texttt{#1}}}
```

da usare come segue:

```
\mail{sit6113@iperbole.%
bologna.it}
```

sit6113@iperbole.bologna.it

3.12 PACCHETTI

Scrivendo un documento, capita non di rado di imbattersi in alcuni problemi che LaTeX non riesce a risolvere da solo. Il suo linguaggio standard, per esempio, non gestisce direttamente l'inclusione delle immagini, né sillaba i testi scritti in lingue diverse dall'inglese, né ancora permette di modificare i margini di pagina in modo semplice. Per aggirare "ostacoli" di questo tipo si sfrutta la struttura modulare del programma, che ne estende le capacità di base: questi moduli aggiuntivi si chiamano *pacchetti*.

Che cosa sono?

Fondamentalmente, un pacchetto è un file "di stile" (con estensione `.sty`) scritto in linguaggio LaTeX, contenente istruzioni che permettono di svolgere alcune operazioni.

Come sapere se servono?

In genere, se per ottenere il risultato sperato si deve faticare troppo, probabilmente qualcuno che si è già trovato nella stessa situazione ha provveduto a creare un pacchetto per semplificare il lavoro.

Viceversa, componendo un file sorgente può capitare che LaTeX produca un messaggio di errore del tipo

```
Can't find file steroid.sty
```

Ciò significa che è stato caricato un pacchetto (steroid, nell'esempio considerato) non presente in TeX Live. In questo caso, bisogna seguire le istruzioni del paragrafo 2.1.2 a pagina 12.

Come scovare il pacchetto che fa al proprio caso?

Questo è l'unico aspetto del lavoro con LaTeX in cui gusto, abilità e fortuna dell'autore la fanno da padroni: cercando sul motore di ricerca del sito del GuIT (☞ GUIT), oppure su CTAN (☞ CTAN) o su Sarovar (☞ SAROVAR), si trovano preziosi riferimenti e soluzioni per risolvere moltissimi problemi.

3.12.1 Come caricarli?

I pacchetti si caricano *nel preambolo* con il comando

```
\usepackage[⟨opzioni⟩]{⟨pacchetto⟩}
```

dove:

- ⟨*opzioni*⟩ è una voce o un elenco di voci separate dalla virgola costituite da un solo elemento o da un'espressione del tipo ⟨*chiave*⟩=⟨*valore*⟩ che specificano le impostazioni del pacchetto;
- ⟨*pacchetto*⟩ è il nome del pacchetto, che va scritto *sempre* con tutte lettere minuscole (LayAureo si scrive così, ma si carica come `layaureo`, per intenderci).

Si noti che si possono caricare più pacchetti *senza opzioni* con uno stesso comando `\usepackage`, separandone i nomi con la virgola.

Tabella 11: Principali unità di misura tipografiche riconosciute da LaTeX (il valore di `em` indicato è valido solo in modo testuale).

Unità	Codice	Valore
centimetro	`cm`	
millimetro	`mm`	
punto tipografico	`pt`	0,3514 mm
x-height	`ex`	Altezza della x nel font in uso
m-width	`em`	Larghezza della M nel font in uso
quadrato	`\quad`	Larghezza della M nel font in uso
quadratone	`\qquad`	Il doppio di `\quad`

3.12.2 Usarli al meglio: la documentazione

Chi scrive o aggiorna un pacchetto per LaTeX ne scrive anche, e quasi sempre in inglese, la *documentazione*. Questa si compone spesso di due parti distinte:

- Il vero e proprio manuale d'uso, che spiega come caricare il pacchetto e usarne i comandi.
- Il codice che costituisce il pacchetto, destinato a chi voglia eventualmente svilupparlo. (Nel caso di pacchetti molto corposi, il codice costituisce un file a sé.)

I pacchetti contenuti in ogni distribuzione di LaTeX sono già corredati della relativa documentazione (quasi sempre un PDF omonimo), facilmente raggiungibile con il programma `texdoc`, integrato in TeX Live. Il programma si lancia dalla riga di comando o con le scorciatoie che di solito ogni editor definisce a questo scopo e prevede numerose opzioni di ricerca, come si può leggere nella sua documentazione: eseguendo

```
texdoc ⟨nome del pacchetto⟩
```

in un attimo si apre il relativo manuale.

3.12.3 E i file `.ins` e `.dtx`?

In alcuni casi, i pacchetti che si scaricano da Internet (in forma di archivi compressi) non contengono direttamente il file `.sty` e il relativo PDF di documentazione, ma un file con estensione `.ins` e uno con estensione `.dtx`. Basta procedere in questo modo:

- Il file `.dtx` contiene la documentazione del pacchetto. Eseguendo LaTeX su di esso, si ottiene il relativo manuale.
- Eseguendo invece LaTeX sul file `.ins`, si ottiene il file `.sty` desiderato (o anche più d'uno, a seconda dei casi).

Si noti che alcuni pacchetti contengono soltanto il file `.dtx`: in questo caso la prima delle due istruzioni permette di ottenere *anche* il file `.sty`.

A questo punto, per poterli usare basta sistemare i file `.sty` ottenuti come descritto nel paragrafo 2.5 a pagina 14.

3.13 UNITÀ DI MISURA TIPOGRAFICHE

Nei prossimi capitoli si useranno spesso istruzioni che richiedono di precisare una lunghezza in una qualche *unità di misura tipografica*. Dal momento

che tali lunghezze spesso poco hanno a che fare con le più conosciute unità di misura del sistema metrico decimale, nella tabella 11 a fronte si mostrano quelle effettivamente usate in questa guida. Si noti che in un ambiente puramente testuale l'uso di `\quad` e `\qquad` è di regola fortemente sconsigliato e va limitato a casi particolarissimi (per spaziare sottooggetti mobili, per esempio, come si spiega nel paragrafo 9.4.2 a pagina 137).

Per maggiori dettagli si veda [Beccari, 2011].

3.14 DOCUMENTI DI GRANDI DIMENSIONI

Per scrivere senza sorprese un documento di grandi dimensioni come un libro o una tesi è importantissimo organizzarne razionalmente il materiale. Immaginando di scrivere la tesi di laurea, si suggerisce di mettere *tutti* i file del documento in una stessa cartella `Tesi`, per esempio, da strutturare in sottocartelle come segue:

- La sottocartella `MaterialeInizialeFinale` contiene il materiale iniziale e finale del documento suddiviso nei corrispondenti file come `Ringraziamenti.tex`, `Introduzione.tex`, `Conclusioni.tex`, eccetera.
- La sottocartella `Capitoli` contiene il materiale principale del documento suddiviso nei corrispondenti file come `Mane.tex`, `Tekel.tex`, `Fares.tex`, eccetera.
- La cartella `Immagini` contiene tutte le immagini da includere nel documento, in uno dei formati accettati da LaTeX. Se sono molte, potrebbe essere conveniente distribuirle in ulteriori sottocartelle corrispondenti ai diversi capitoli. Immaginando di chiamarle `grafici` e `foto`, basta scrivere nel file di impostazioni (si veda poco più sotto)

```
\graphicspath{{grafici/},{foto/}}
```

La cartella `Tesi` *deve* contenere anche altri due file:

- il file principale del documento, `Tesi.tex`,
- il file `Bibliografia.bib` (a meno di non volerlo sistemare come spiegato nel paragrafo 7.2 a pagina 111)

Si consiglia infine di raccogliere le definizioni di comandi e ambienti ad hoc in un pacchetto personale `impostazioni-tesi.sty` (si scrive con l'editor in uso, si registra con estensione `.sty` e *non* va composto), da caricare nel preambolo come un normale pacchetto immediatamente prima dell'inizio del documento. Questi piccoli accorgimenti "puliscono" il file principale semplificando notevolmente il proprio lavoro.

I file `.tex` delle varie sezioni si caricano poi uno dopo l'altro nell'ambiente `document` semplicemente indicandone il percorso nell'argomento del comando `\input`, come mostra l'esempio seguente:

```
\documentclass[⟨...⟩]{⟨...⟩}
\usepackage{⟨...⟩}
...
\usepackage{impostazioni-tesi}

\begin{document}

...
\input{MaterialeInizialeFinale/Ringraziamenti}
```

```
\input{MaterialeInizialeFinale/Introduzione}
\input{Capitoli/Mane}
\input{Capitoli/Tekel}
\input{Capitoli/Fares}
...
\input{MaterialeInizialeFinale/Conclusioni}
...

\end{document}
```

In pratica, `\input` costruisce il documento “attaccando” uno dopo l’altro i vari file sorgente in cui il documento è stato suddiviso. Si noti che:

- questi ultimi *non* devono contenere alcun preambolo (da scrivere invece *una sola volta* nel file principale), ma soltanto titolo della sezione e relativo contenuto;
- `\input` permette l’annidamento (cioè ammette altri `\input` nel proprio argomento);
- per evitare problemi, è opportuno che il percorso dei file inclusi mediante i comandi `\input` e `\graphicspath` sia *relativo* (un percorso assoluto, infatti, verrebbe addirittura *rifiutato* da LaTeX per motivi di sicurezza) e *non* contenga spazi.

4 TESTO

Questo capitolo descrive gli strumenti fondamentali definiti da LaTeX per il trattamento del testo.

4.1 STRUTTURA DEL TESTO

Lo scopo principale di chi scrive un testo è comunicare idee e conoscenze al lettore, che le comprenderà tanto più quanto meglio sono strutturate, e ne apprezzerà tanto più la struttura quanto più la forma tipografica del documento rispecchia la costruzione logica del suo contenuto [Oetiker *et al.*, 2011].

La struttura interna di un'opera non letteraria di grandi dimensioni presenta quasi sempre due elementi costanti:

- la suddivisione in "blocchi" di testo di lunghezza variabile, con i blocchi più grandi che contengono quelli più piccoli;
- la presenza di altre parti aggiunte (come l'introduzione e le eventuali appendici).

Le suddivisioni elencate nella tabella 12 sono fondamentali per comprendere l'articolazione di un testo scritto, e vengono chiamate in generale *sezioni*.

A questo proposito si noti che:

- il sezionamento del documento rimane a carico dell'autore, perché LaTeX non lo fa automaticamente;
- è di fondamentale importanza articolare il testo in capoversi, sia per chi scrive sia per chi legge: le informazioni sono meglio articolate e più facilmente memorizzabili.

4.2 COMPORRE I CAPOVERSI

Spesso si sottovaluta l'importanza di scrivere un testo ben strutturato, e usando LaTeX altrettanto frequentemente si comincia un nuovo capoverso senza nemmeno rendersene conto.

È molto facile commettere quest'ultimo errore se il testo contiene formule matematiche [Oetiker *et al.*, 2011]. Infatti l'abitudine, diffusa, di lasciare una riga vuota tra la fine di una formula e la prosecuzione del testo si sconta nel documento finito con altrettanti nuovi capoversi, anche là dove il flusso del discorso non li richiederebbe per nulla.

Cominciare un nuovo capoverso

Con l'ovvia eccezione del primo capoverso di una sezione, per cominciare un nuovo capoverso con LaTeX si hanno due possibilità:

- si lascia una riga vuota nel codice sorgente (di solito si fa così);
- si dà il comando `\par`.

Tabella 12: Lunghezza orientativa delle sezioni di un testo scritto.

Sezione	Lunghezza orientativa
Parte	Imprecisabile
Capitolo	Da una decina a un centinaio di pagine
Paragrafo	Da mezza a una decina di pagine
Sottoparagrafo	Da poche righe a un paio di pagine
Capoverso	Da una a una ventina di righe
Enunciato	Da una parola a una decina di righe

In ogni caso, non si cominci *mai* un nuovo capoverso con \\ (in qualche esempio di questa guida lo si è fatto per motivi di spazio).

Osservando gli esempi che seguono, si cerchi di capire perché a volte c'è la riga bianca e altre no. (Se non si comprendono ancora tutti i comandi, si leggano interamente questo capitolo e i primi paragrafi del capitolo 5 a pagina 59, e poi si ritorni su questo punto.)

```
\dots quando Einstein propose
l'equazione
\begin{equation}
  E = mc^2,
\end{equation}
che è allo stesso tempo la più
nota e la meno compresa
formula della Fisica.
```

. . . quando Einstein propose l'equazione

$$E = mc^2, \tag{4.1}$$

che è allo stesso tempo la più nota e la meno compresa formula della Fisica.

```
\dots che, rispetto ai
precedenti, presenta alcuni
vantaggi.

La formula
\begin{equation}
  I_\textup{D} =
  I_\textup{F} - I_\textup{R}
\end{equation}
costituisce la parte centrale
di un modello molto
diverso di transistor. \dots
```

. . . che, rispetto ai precedenti, presenta alcuni vantaggi.

La formula

$$I_D = I_F - I_R \tag{4.2}$$

costituisce la parte centrale di un modello molto diverso di transistor. . . .

```
\dots da cui segue la legge di
Kirchhoff sulle correnti:
\begin{equation}
  \sum_{k=1}^n I_k = 0.
\end{equation}

La legge di Kirchhoff sulle
tensioni può essere
ricavata\dots
```

. . . da cui segue la legge di Kirchhoff sulle correnti:

$$\sum_{k=1}^{n} I_k = 0. \tag{4.3}$$

La legge di Kirchhoff sulle tensioni può essere ricavata. . .

Terminare un capoverso con una formula in display (si veda il paragrafo 5.1 a pagina 59) come nel terzo degli esempi proposti è raro, ma comunque lecito. È invece *sempre* sconsigliabile cominciarlo con una formula matematica di qualunque tipo.

Capoversi ben composti e microtipografia

Un documento "ben composto" si riconosce da alcuni elementi: il testo è giustificato, le righe sono interamente riempite, le parole sono adeguatamente spaziate tra loro e sillabate a fine riga se proprio non ci stanno. I capoversi presentano la prima riga rientrata per facilitare la lettura e non sono solcati dai fastidiosi "ruscelli" (le zone bianche che percorrono la pagina in verticale). Di solito tutto questo si ottiene dando a mano le rispettive impostazioni; con LaTeX, invece, non occorre nemmeno pensarci, perché il programma:

- giustifica il testo per impostazione predefinita;
- riempie la riga con un sofisticatissimo algoritmo di spaziatura tra le parole, sillabandole solo se *davvero* necessario;
- rientra automaticamente la prima riga di ogni capoverso tranne il primo (se per qualche motivo non si volesse il rientro, basta cominciare la riga interessata con `\noindent`);
- numera automaticamente le pagine del documento;
- non aggiunge spazio supplementare tra un capoverso e l'altro.

Talvolta, invece, questo spazio supplementare potrebbe servire. Lo si può inserire con i seguenti comandi:

- `\bigskip`, `\medskip` e `\smallskip` inseriscono uno spazio verticale rispettivamente "grande", "medio" e "piccolo" la cui ampiezza è in funzione del font utilizzato.
- `\vspace{`⟨*lunghezza*⟩`}` inserisce uno spazio verticale pari a ⟨*lunghezza*⟩ (che va perso se dopo la composizione viene a trovarsi all'inizio di una pagina: per mantenerlo basta usare la forma `\vspace*`.)

La tipografia anglosassone (predefinita in LaTeX) non prevede il rientro della prima riga del primo capoverso di una sezione (si veda il paragrafo 3.7 a pagina 31). Per ottenerlo, secondo una consuetudine spesso seguita in Italia, basta semplicemente caricare il pacchetto indentfirst nel modo consueto.

Il pacchetto microtype migliora il riempimento delle righe abilitando:

- l'*espansione dei font*, ovvero espande i caratteri per riempire la riga in modo ottimale;
- la *protrusione dei caratteri*, ovvero permette a certi caratteri di sporgere leggerissimamente a fine riga (di solito segni di punteggiatura e trattini).

Perciò si consiglia di caricarlo *sempre*.

Interrompere una riga senza cominciare un nuovo capoverso

In casi particolari può essere necessario interrompere una riga. Per farlo si usano i comandi `\\` o `\newline`, e
se ne incomincia una nuova *senza iniziare un nuovo capoverso* (e senza rientro, dunque, come qui).

Si può inserire uno spazio aggiuntivo tra due linee dello stesso capoverso con il comando `\\[`⟨*lunghezza*⟩`]`, in cui ⟨*lunghezza*⟩ può essere espressa in una qualunque delle unità di misura tipografiche accettate da LaTeX avendo cura di usare *il punto* come separatore decimale (si veda la tabella 11 a pagina 40).

Dividere le parole a fine riga

In generale, LaTeX cerca di interrompere le righe sempre nel miglior punto possibile. Se, però, non riesce a farlo neppure secondo i suoi severi criteri, le lascia fuoriuscire dal margine destro e avverte l'autore con un messaggio di *overfull hbox*. Non sempre è facile individuarle: nel capitolo 10 a pagina 143 si spiega come fare.

L'algoritmo di sillabazione di LaTeX funziona correttamente con quasi tutte le parole, ma in particolari circostanze si potrebbe volere una divisione diversa da quella automatica. Con i nomi propri o i tecnicismi come *nitroidrossilamminico* o *macroistruzione*, per esempio, a volte si richiede la sillabazione etimologica anziché quella che LaTeX esegue di default: "nitro-idrossil-amminico" invece di "ni-troi-dros-si-lam-mi-ni-co" e "ma-cro-istru-zio-ne" invece di "ma-croi-stru-zio-ne".

In questi casi basta scrivere le parole nell'argomento di `\hyphenation` (nel preambolo) *già sillabate*, separandole con uno spazio ed evitando caratteri speciali e simboli:

```
\hyphenation{nitro-idrossil-amminico ma-cro-istru-zio-ne}
```

Il comando precedente funziona anche al contrario. Una scrittura come la seguente

```
\hyphenation{nitro-idrossil-amminico FORTRAN}
```

sillaba *nitroidrossilamminico* e *Nitroidrossilamminico* come suggerito nell'argomento, ma non FORTRAN, *Fortran* e *fortran*. Analogamente, il comando `\hyphenation` si può usare per forzare nella parola qualunque cesura si desideri: se si vuole spezzare la parola *melograno* soltanto tra *melo* e *grano*, si scrive:

```
\hyphenation{melo-grano}
```

Se la parola in questione compare nel documento una sola volta, si può suggerirne la sillabazione direttamente nel testo. Il comando `\-` spezza la parola nel punto (o nei punti) in cui viene dato, e *in quel punto soltanto*.

```
La scoperta dell'acido
nitro\-idrossil\-amminico
avvenne nel 1896.
```

La scoperta dell'acido nitroidrossil-amminico avvenne nel 1896.

Si noti che anche gli interventi sulla sillabazione, come tutti quelli operati "a mano" sul documento, dovrebbero essere effettuati durante la revisione finale immediatamente precedente la stampa. Di fronte a un messaggio di *overfull hbox*, per esempio, è *sempre* preferibile tentare di riformulare l'enunciato piuttosto che imporre una sillabazione particolare come prima soluzione al problema.

L'opzione `italian` di babel definisce il comando `"/`, utile per andare a capo dopo la barra nelle espressioni che comportano l'alternanza tra due possibilità. La scrittura `modulazione"/demodulazione`, per esempio, produce *modulazione/demodulazione* se si trova all'interno di una riga, *modulazione/ demodulazione* se invece si trova alla fine.

Il comando

```
\mbox{⟨testo⟩}
```

serve per mantenere unita una parola *senza* usare `\hyphenation`. Va usato all'occorrenza, magari perché in un certo punto del documento non va bene che la parola sia spezzata, ma altrove sì:

```
Entro quest'anno avrò imparato il
Fortran. \\
Entro quest'anno avrò imparato il
\mbox{Fortran}.
```

Entro quest'anno avrò imparato il Fortran.
Entro quest'anno avrò imparato il Fortran.

Spazio tra le parole

Per ottenere capoversi giustificati LaTeX inserisce tra le parole spazi di dimensione variabile e, per rendere il testo più leggibile, dopo un punto di fine enunciato inserisce uno spazio leggermente più ampio di quanto farebbe un comune elaboratore di testo. Il programma, infatti, presuppone che gli enunciati si chiudano con un punto, un punto interrogativo o un punto esclamativo.

Non tutti i punti, però, sono uguali, per cui le eccezioni alle regole generali appena esposte devono essere specificate dall'autore. I casi sono tre:

- immediatamente *dopo* il punto che chiude un'abbreviazione a metà enunciato (tranne se l'abbreviazione è l'ultima parola), si usa la sequenza `\␣`;
- immediatamente *prima* di un punto di fine enunciato che segue una maiuscola (che LaTeX interpreta comunque come un punto che chiude un'abbreviazione, omettendo lo spazio supplementare) si usa il comando `\@`;
- la tilde ~ produce uno *spazio indivisibile*, utile per mantenere unite espressioni che non possono *mai* essere divise da un fine riga.

L'esempio seguente mostra `\@` all'opera:

```
CEE. Poi CE. Ora UE. \\
CEE\@. Poi CE\@. Ora UE\@.
```

CEE. Poi CE. Ora UE.
CEE. Poi CE. Ora UE.

Come si può osservare, la spaziatura corretta si ottiene con la seconda scrittura. Per disabilitare lo spazio aggiuntivo dopo un punto fermo *in tutto* il documento (come nella tipografia francese), si dà nel preambolo la dichiarazione `\frenchspacing`. Se si vuole comporre in questo modo soltanto alcune parti del lavoro, la si può dare direttamente nel corpo principale del testo (in tal caso, non serve più `\@` ogni volta).

Si osservi come agisce la tilde nei due esempi seguenti:

```
Avevamo letto le dispense del
prof. Beccari. \\
Avevamo letto le dispense del
prof.~Beccari.
```

Avevamo letto le dispense del prof. Beccari.
Avevamo letto le dispense del prof. Beccari.

```
Il concetto è spiegato nel
paragrafo \ref{sec:par}. \\
Il concetto è spiegato nel
paragrafo~\ref{sec:par}.
```

Il concetto è spiegato nel paragrafo 4.2.
Il concetto è spiegato nel paragrafo 4.2.

Come si può notare, per evitare che una riga finisca con un punto non di fine enunciato o cominci con un numero, le scritture corrette sono le seconde di ciascuna coppia.

Si noti che LaTeX *non* mette automaticamente in maiuscolo l'iniziale della prima parola dopo un punto (di qualunque specie sia), a differenza di quanto accade con la maggior parte degli altri programmi di videoscrittura.

Tabella 13: Virgolette, tratti e puntini di sospensione. Per evidenziare le differenze tra i vari segni, virgolette alte e apici sono composti con il font Computer Modern.

Segno		Codice	Risultato
Virgolette	semplici alte	` '	‘ ’
	doppie alte	`` ''	“ ”
		“ ”	“ ”
	doppie basse	<< >>	« »
		« »	« »
Tratti	trattino	-	-
	tratto	--	–
	lineetta	---	—
	meno	$-$	−
Puntini		\dots	...

4.3 CARATTERI PARTICOLARI E SIMBOLI

4.3.1 Virgolette, tratti e puntini di sospensione

Virgolette

In tipografia si usano comunemente tre tipi di virgolette: gli ‘apici’, le “virgolette inglesi” e le «virgolette caporali». I modi per ottenerle sono molti, e dipendono dalla codifica di input impostata, dall’editor in uso e dalle particolari sequenze di tasti digitate: la tabella 13 raccoglie quelli più usati (si lascia all’autore scoprire le alternative).

Gli esempi seguenti le mostrano all’opera:

```
Ora dovrebbe essere chiaro
il concetto di
‘composizione asincrona’.
```

Ora dovrebbe essere chiaro il concetto di ‘composizione asincrona’.

```
La Delta di Dirac è una
‘‘funzione impropria’’.
```

La Delta di Dirac è una “funzione impropria”.

```
<<Se stavi attento, Ermanno,
capivi tutto anche tu.>>
```

«Se stavi attento, Ermanno, capivi tutto anche tu.»

Si noti che il font principale di questa guida restituisce le forme aperta e chiusa di virgolette inglesi e apici in modo molto simile (ciò non accade con altri font).

Trattini, tratti e lineette

La tipografia distingue quattro tipi di tratto: tre (*trattino*, *tratto* e *lineetta*) corrispondono a un diverso numero di trattini consecutivi, mentre il quarto è il segno matematico *meno*. La tabella 13 mostra come ottenerli, e gli esempi seguenti ne illustrano alcuni possibili usi:

```
Stratford-on-Avon, e-mail \\
p.~13-67, 1921-28 \\
Ottica~---~Schema generale \\
---\,Eccomi\,---~disse. \\
$0$, $1$ e $-1$
```

Stratford-on-Avon, e-mail
p. 13-67, 1921-28
Ottica – Schema generale
— Eccomi — disse.
0, 1 e -1

Tabella 14: Loghi d'uso frequente.

Comando	Risultato	Pacchetto richiesto
`\TeX`	TeX	
`\LaTeX`	LaTeX	
`\LaTeXe`	LaTeX 2ε	
`\AmS`	AMS	amsmath
`\MF`	METAFONT	mflogo
`\MP`	METAPOST	mflogo
`\GuIT`, `\GuIT*`	GuIT, GuIT	guit
`\Ars`	ArsTeXnica	guit

Tabella 15: Accenti e caratteri particolari.

`\'o`	ò	`\u{o}`	ŏ	`\~o`	õ	`\.o`	ȯ
`\'o`	ó	`\t{oo}`	o͡o	`\r{o}`	o̊	`\d{o}`	ọ
`\^o`	ô	`\"o`	ö	`\c{o}`	o̧	`\=o`	ō
`\v{o}`	ǒ	`\H{o}`	ő	`\k{o}`	ǫ	`\b{o}`	o̱
`\OE`	Œ	`\AE`	Æ	`\AA`	Å	`\O`	Ø
`\oe`	œ	`\ae`	æ	`\aa`	å	`\o`	ø
`\L`	Ł	`\DH`	Ð	`\DJ`	Đ	`\TH`	Þ
`\l`	ł	`\dh`	ð	`\dj`	đ	`\th`	þ

Puntini di sospensione e segno di omissione

Se inseriti battendo tre punti consecutivi, i *puntini di sospensione* potrebbero compromettere la spaziatura delle parole o la corretta interruzione di riga. LaTeX risolve il problema definendo il comando `\dots`, che li stampa correttamente spaziati e li tiene uniti *in ogni caso*:

```
Non così... ma così:
Londra, Parigi\dots{} Berlino.
```

Non così... ma così: Londra, Parigi... Berlino.

Per indicare l'omissione di una parola o una porzione di testo originali, si può usare il segno di *omissione* [...] definendo nel preambolo un comando ad hoc `\omissis` (vedi il paragrafo 9.1 a pagina 127):

```
\newcommand{\omissis}{[\textellipsis\unkern]}
```

Si noti che `\dots` e `\omissis` sono comandi "che producono testo": vanno separati dalla parola seguente con uno spazio esplicito nei modi consueti.

4.3.2 Loghi, accenti, caratteri particolari, apici e pedici

Loghi, accenti e caratteri particolari

La tabella 14 elenca i comandi per stampare i loghi più comuni.

LaTeX permette di usare gli accenti e i caratteri particolari di molte lingue (nella tabella 15 sono esemplificati per la lettera *o*, ma funzionano anche per tutte le altre lettere), come si può vedere nell'esempio seguente:

```
H\^otel, na\"if,  Stra\ss e,
!`Se\~norita!, élève,
Sm\o rrebr\o d, z\l oty.
```

Hôtel, naïf, Straße, ¡Señorita!, élève, Smørrebrød, złoty.

Il simbolo dell'euro (€) si ottiene con il comando `\euro` del pacchetto eurosym.

Apici e pedici

L'opzione `italian` di `babel` definisce una coppia di comandi che stampano il loro argomento in tondo *anche in modo matematico*. In modo testuale, inoltre, mantengono anche lo stile corrente, qualunque sia:

- `\ap{⟨testo⟩}` stampa un apice come nelle abbreviazioni oggi in disuso *sig.*^{ra} o *f.*^{lli} (l'alternativa è `\textsuperscript{⟨testo⟩}`);
- `\ped{⟨testo⟩}` stampa un pedice, utile per qualche sostanza chimica come la vitamina B_{12}, per esempio.

Per l'elenco completo di *tutti* i simboli e i caratteri speciali di LaTeX (quasi 6000), si veda [Pakin, 2009].

4.4 TITOLI E FRONTESPIZI

Titoli standard

Il comando

```
\maketitle
```

dato *dopo* `\begin{document}` stampa il titolo del documento. Gli elementi che lo costituiscono vengono definiti dai comandi

```
\title{⟨titolo⟩}
\author{⟨autore⟩}
\date{⟨data⟩}
```

Il loro funzionamento si spiega da sé, ma si osservi quanto segue:

- un ⟨*titolo*⟩ troppo lungo per stare su una sola riga si spezza con `\\` (ma lo si eviti per quanto possibile);
- i vari ⟨*autore*⟩ di un documento scritto a più mani si separano con `\and`;
- LaTeX stampa come predefinita la ⟨*data*⟩ della composizione anche se `\date` non viene dato, mentre la omette se si lascia vuoto il suo argomento (`\date{}`).

Per inserire ringraziamenti veloci si usa il comando

```
\thanks{⟨ringraziamenti⟩}
```

che nelle classi standard e produce il proprio argomento come una nota al piede con un simbolo in esponente. Si può dare il comando *dentro* l'argomento di uno qualunque dei tre comandi appena esaminati. Ecco un esempio:

```
\author{Lorenzo Pantieri \and Tommaso Gordini%
       \thanks{Ringraziamo i membri del \GuIT.}}
```

Nelle classi $\mathcal{AMS}$ `\thanks` va invece dato in una riga a sé, *fuori* dai tre comandi appena considerati.

Frontespizio

Il titolo generato dal comando `\maketitle`, si deve riconoscere, è piuttosto spartano, anche se in articoli e relazioni può essere accettabile. Si consiglia di comporre il frontespizio di una tesi di laurea o di dottorato con

Figura 6: Esempio d'uso di frontespizio.

il pacchetto frontespizio (si veda la figura 6), accompagnato da un'esauriente documentazione (in italiano). Oltre ai comandi per inserire tutti i dati che di solito trovano posto in una pagina di questo tipo, il pacchetto prevede opzioni per l'uso di caratteri senza grazie e l'inserimento di loghi universitari e immagini in filigrana.

In alternativa si può usare la *suite* ClassicThesis, che comprende un modello di tesi pronto per l'uso completo di frontespizio [Pantieri, 2007].

Infine, se proprio nessuna delle soluzioni precedenti va bene, si può comporre un frontespizio personalizzato con l'ambiente `titlepage` (da aprire *subito dopo* `\begin{document}`) all'interno del quale si è completamente padroni dell'impaginazione.

4.5 NOTE A MARGINE E A PIÈ DI PAGINA

In linea generale si usino le note *con grande moderazione*: specie quelle al piede, infatti, interrompono la lettura e possono dare seri problemi d'impaginazione. Si tenga presente che una nota *deve* potersi omettere leggendo: se il suo contenuto si rivela essenziale per la comprensione, va evidentemente levato dalla nota e messo nel corpo del testo.

Note a margine

Una nota di questo tipo si ottiene molto semplicemente con il comando

```
\marginpar{⟨testo della nota a margine⟩}
```

Nei documenti impostati per la stampa in fronte/retro le note vengono stampate nel margine destro nelle pagine dispari e nel margine sinistro in quelle pari. Nei documenti solo fronte, invece, le note vengono stampate sempre nel margine destro. (Talvolta può accadere che una nota appaia nel margine sbagliato: per rimediare a questo piccolo difetto di LaTeX basta caricare il pacchetto mparhack.)

Note a piè di pagina

Il comando

```
\footnote{⟨testo della nota a piè di pagina⟩}
```

produce una nota in fondo alla pagina corrente con un riferimento nel testo costituito da un numero in esponente. Le note al piede dovrebbero essere messe, per quanto possibile, alla fine del relativo capoverso, *dopo* il punto fermo.[1]

```
Le note a piè di pagina sono
l'emblema della meticolosità.%
\footnote{Eccone un esempio.}
```

Le note a piè di pagina sono l'emblema della meticolosità.[a]

[a] Eccone un esempio.

Si tenga presente che:

- la numerazione delle note riprende a ogni `\chapter` o `\section`;
- se in un documento le note sono poche o pochissime, potrebbe essere meglio inserirle con un simbolo in esponente anziché il numero predefinito: basta scrivere nel preambolo

  ```
  \renewcommand{\thefootnote}{\fnsymbol{footnote}}
  ```

LaTeX produce note di alta qualità, ma quando circostanze particolari le richiedono in un formato diverso da quello predefinito basta caricare il pacchetto footmisc e configurarlo opportunamente (se ne veda la documentazione).

4.6 EVIDENZIARE LE PAROLE

Scrivendo a macchina, le parole importanti `si evidenziano` <u>`con`</u> <u>`una`</u> <u>`sottolineatura`</u>; in tipografia, invece, le parole si evidenziano mettendole *in corsivo*. Le possibilità sono due.

Per evidenziare una parola o una porzione di testo *indipendentemente dal contesto* in cui si trovano, LaTeX definisce il comando standard

```
\emph{⟨testo⟩}
```

che si vede all'opera nell'esempio seguente:

```
\emph{All'interno di un testo
già evidenziato, \LaTeX{}
evidenzia ulteriormente con il
\emph{font tondo}.}
```

All'interno di un testo già evidenziato, LaTeX evidenzia ulteriormente con il font tondo.

Esiste anche un altro comando standard, `\textit`, che produce il proprio argomento in corsivo *in ogni caso*. Per cogliere la differenza logica tra corsivo ed evidenziato, si osservino le due scritture:

```
\emph{Sarò lì in
\textit{dieci} minuti.} \\
\textit{Sarò lì in
\emph{dieci} minuti.}
```

Sarò lì in dieci minuti.
Sarò lì in dieci *minuti.*

1 Così.

4.7 AMBIENTI TESTUALI

Le sezioni seguenti presentano i principali ambienti standard che LaTeX definisce per trattare il testo.

4.7.1 Elenchi puntati, numerati e descrizioni

In un documento gli elenchi sono molto importanti. Permettono, infatti:

- di far "respirare" il testo;
- di migliorarne la leggibilità;
- di strutturare il pensiero.

L'elenco precedente è stato ottenuto con l'ambiente `itemize` come segue:

```
Gli elenchi permettono:
\begin{itemize}
\item di far ‘‘respirare’’
      il testo;
\item di migliorarne
      la leggibilità;
\item di strutturare il pensiero.
\end{itemize}
```

Gli elenchi permettono:

- di far "respirare" il testo;
- di migliorarne la leggibilità;
- di strutturare il pensiero.

Il comando `\item` stampa un pallino nero prima di ogni elemento dell'elenco.

L'ambiente `enumerate` si usa come `itemize`, ma qui a ogni elemento `\item` premette un numero puntato:

```
Ecco un elenco numerato:
\begin{enumerate}
\item Mane;
\item Tekel;
\item Fares.
\end{enumerate}
```

Ecco un elenco numerato:

1. Mane;
2. Tekel;
3. Fares.

Si noti che è opportuno usare una lista numerata se in seguito ci si deve riferire a un suo elemento particolare (anche assegnandole un'etichetta) o se per esempio si devono descrivere le fasi di un procedimento. In caso contrario, è più opportuna una lista puntata.

L'ambiente `description` si usa per le *descrizioni*, elenchi in cui il segno distintivo è una parola o un'espressione che s'intende descrivere o spiegare, da scrivere nell'argomento facoltativo (*in questo caso*, però, obbligatorio) di `\item`:

```
E ora una descrizione:
\begin{description}
\item[itemize] Per gli
     elenchi puntati.
\item[enumerate] Per gli
     elenchi numerati.
\item[description] Per gli
     elenchi in cui ogni
     elemento comincia con
     un testo a piacere.
\end{description}
```

E ora una descrizione:

ITEMIZE Per gli elenchi puntati.

ENUMERATE Per gli elenchi numerati.

DESCRIPTION Per gli elenchi in cui ogni elemento comincia con un testo a piacere.

Come si può osservare, LaTeX evidenzia automaticamente l'argomento del comando in accordo con le impostazioni generali della classe di documento in uso.

LaTeX permette l'annidamento degli ambienti (e dunque anche degli elenchi), purché l'ordine di chiamata venga rispettato:

```
\begin{⟨aaa⟩}
...
\begin{⟨bbb⟩}
...
\end{⟨bbb⟩}
...
\end{⟨aaa⟩}
```

Ecco un esempio di elenchi annidati (si consiglia di non annidare mai più di una lista dentro l'altra):

```
Gli elenchi:
\begin{itemize}
\item sono facili da usare;
\item rendono più chiaro
      il testo:
      \begin{itemize}
      \item spaziandolo;
      \item facilitandone
            la lettura;
      \end{itemize}
\item permettono di strutturare
      meglio il pensiero.
\end{itemize}
```

Gli elenchi:

- sono facili da usare;
- rendono più chiaro il testo:
  - spaziandolo;
  - facilitandone la lettura;
- permettono di strutturare meglio il pensiero.

Si noti che LaTeX provvede automaticamente a cambiare il contrassegno delle voci degli elenchi annidati, che si possono individuare più facilmente nel testo sorgente rientrandoli leggermente (può essere una buona abitudine, ma non è una regola).

Di seguito si riportano alcune convenzioni tipografiche comunemente seguite per comporre gli elenchi (sono le stesse che si sono osservate in questa guida):

- ogni elemento di un elenco *semplice* (i cui elementi sono costituiti da un solo enunciato) comincia di solito con l'iniziale minuscola della prima parola e termina con un punto e virgola tranne l'ultimo, che termina con un punto fermo;

- ogni elemento di un elenco *complesso* (in cui *almeno uno* degli elementi sia composto da più di un enunciato) comincia di solito con l'iniziale maiuscola della prima parola (anche dopo il segno di due punti) e termina con il punto fermo.

Non bisogna per forza uniformare *tutti* gli elenchi di un documento a criteri stabiliti a priori: l'importante è essere coerenti volta per volta.

Per personalizzare gli elenchi in ogni loro aspetto c'è il pacchetto `enumitem` (se ne veda la documentazione).

4.7.2 Allineare e centrare i capoversi

LaTeX definisce tre ambienti standard per allineare un capoverso a sinistra:

```
\begin{flushleft}
Questo testo è allineato a \\
sinistra. \LaTeX{} non cerca
di creare righe di uguale
lunghezza.
\end{flushleft}
```

Questo testo è allineato a
sinistra. LaTeX non cerca di creare
righe di uguale lunghezza.

a destra:

```
\begin{flushright}
Questo testo è allineato a \\
destra. \LaTeX{} non cerca
di creare righe di uguale
lunghezza.
\end{flushright}
```

Questo testo è allineato a
destra. LaTeX non cerca di creare
righe di uguale lunghezza.

o per centrarlo sulla pagina:

```
\begin{center}
Al centro \\
dell'universo.
\end{center}
```

Al centro
dell'universo.

Come si può osservare, le righe di testo vanno a capo automaticamente, a meno di un'interruzione esplicita con `\\`.

Dedica

Non esistono regole tipografiche vincolanti per stampare la dedica in una pubblicazione, se non il gusto e le esigenze dell'autore. In linea generale, se di dimensioni contenute si colloca orizzontalmente sulla pagina con uno degli ambienti appena considerati.

Altrettanto libero ne è il collocamento in verticale, che si può controllare con i comandi opportuni: qualche prova permette di ottenere il risultato desiderato.

4.7.3 Citazioni e intercitazioni

Esistono due modi per scrivere le citazioni con LaTeX: "in linea" e "in display".

Citazioni in linea

Una citazione "in linea" è un testo tra virgolette appartenente al flusso del discorso, come quando si cita il motto kantiano «il cielo stellato sopra di me, la legge morale dentro di me».

```
Una citazione in linea
è un'espressione appartenente
al flusso del discorso:
«il cielo stellato sopra di me,
la legge morale dentro di me».
```

Una citazione in linea è un'espressione appartenente al flusso del discorso: «il cielo stellato sopra di me, la legge morale dentro di me».

Citazioni di questo tipo sono generalmente di proporzioni contenute.

Citazioni in display

Una citazione "in display" è un testo che va composto entro margini più ampi di quelli in uso e separandolo da quanto precede e segue con spazi bianchi di ampiezza adeguata, in modo che si trovi "in mostra" e bene in risalto sulla pagina.

I due ambienti standard definiti da LaTeX allo scopo, `quote` e `quotation`, non sono del tutto soddisfacenti perché, per esempio, non riducono automaticamente il corpo della citazione come richiedono le buone tradizioni tipografiche. Per ottenere citazioni in display ben composte si consiglia il pacchetto quoting, da impostare come segue nel preambolo se si prevede di mantenere lo stesso stile in tutte le citazioni del documento (scelta consigliata):

```
\usepackage{quoting}
\quotingsetup{font=small}
```

Il pacchetto definisce l'omonimo ambiente `quoting` da usare come segue:

```
Una citazione in display è
un testo che \LaTeX{} compone
su linee a sé:
\begin{quoting}
Il cielo stellato sopra di me,
la legge morale dentro di me.
\end{quoting}
Come si può osservare, la
citazione è centrata e
separata dal resto del testo.
```

Una citazione in display è un testo che LaTeX compone su linee a sé:

> Il cielo stellato sopra di me, la legge morale dentro di me.

Come si può osservare, la citazione è centrata e separata dal resto del testo.

Si noti che la citazione può essere lunga a piacere e suddivisa in capoversi, se necessario. In particolare:

- se la citazione continua il discorso principale non deve avere la prima riga rientrata: in questo caso nel testo sorgente `quoting` seguirà immediatamente il testo che precede *senza righe bianche in mezzo*;
- se la citazione comincia un nuovo capoverso, invece, il rientro ci va: basta lasciare una riga bianca tra l'ambiente e il testo precedente.

In entrambi i casi, la prima riga dei capoversi successivi al primo viene rientrata per impostazione predefinita.

Per ulteriori esigenze (citazioni in lingua straniera, indicazione della fonte, eccetera) si rimanda al paragrafo 9.2.3 a pagina 133 e al pacchetto csquotes.

Citazioni annidate

Per le *citazioni annidate* (cioè citazioni dentro altre citazioni) le convenzioni oscillano, per cui la cosa migliore è decidere per l'una o l'altra soluzione e mantenerla in tutto il documento. In generale:

- se è in linea, si può mettere la citazione tra virgolette diverse da quelle scelte per la citazione principale;
- se è in display, la si può mettere tra virgolette oppure in un altro ambiente `quoting` annidato in quello principale.

4.7.4 Poesie

Per scrivere poesie LaTeX definisce l'ambiente standard `verse`, nel quale:

- i margini vengono aumentati come per le citazioni;
- ogni verso, tranne l'ultimo della strofa, *deve* finire con `\\`;
- le strofe vengono separate automaticamente tra loro con dello spazio bianco.

L'esempio seguente mostra l'ambiente all'opera:

```
La poesia \emph{Un'altra notte}
è di Giuseppe Ungaretti.
\begin{verse}
In quest'oscuro \\
colle mani \\
gelate \\
distinguo \\
il mio viso

Mi vedo \\
abbandonato nell'infinito
\end{verse}
```

La poesia *Un'altra notte* è di Giuseppe Ungaretti.

In quest'oscuro
colle mani
gelate
distinguo
il mio viso

Mi vedo
abbandonato nell'infinito

Per esigenze poetiche più avanzate è utile il pacchetto verse (se ne veda la documentazione).

4.7.5 Codici e algoritmi

Talvolta bisogna scrivere nel documento frammenti o "pezzi" di testo più consistenti in modo *verbatim* ("alla lettera"), cioè mettendo in evidenza il linguaggio di programmazione con cui sono stati scritti, nel quale spazi, caratteri speciali, rientri, a capo, simboli e comandi possono avere un'importanza particolare e *non* devono essere interpretati. In questa guida lo si è fatto tutte le volte che si è mostrato il codice di un comando, di un ambiente o quello per ottenere un certo risultato in stampa, come per la poesia precedente.

Per scrivere un frammento di testo verbatim *in linea* e che non debba andare a capo, LaTeX definisce il comando standard

```
\verb!⟨testo verbatim⟩!
```

Il carattere `!` è solo uno dei possibili *caratteri delimitatori*, cioè caratteri che hanno la sola funzione di indicare l'inizio e la fine del ⟨*testo verbatim*⟩: a questo scopo si può usare un carattere qualunque, *tranne* `*`, purché non compaia tra i caratteri da riprodurre. Si consiglia di usarne uno tra `! ? | @`. Si noti che il ⟨*testo verbatim*⟩ *non* obbedisce ai parametri di giustificazione attivi nel documento: se troppo lungo, infatti, fuoriesce dal margine destro.

Per scrivere del testo verbatim *in display e su più righe*, invece, c'è l'ambiente standard `verbatim`, da usare come di consueto. A questo proposito si noti che:

- né `\verb` né `verbatim` possono comparire nell'argomento di un altro comando;
- `verbatim` può riprodurre tutto *tranne* il comando `\end{verbatim}` (davanti a un ambiente chiuso due volte, infatti, il compositore si arresta restituendo un errore).

Sia \verb sia verbatim prevedono una variante asterisco che riproduce lo spazio in modo visibile con il carattere ␣, come si può osservare negli esempi seguenti.

```
Il logo ``\LaTeX'' si ottiene
con il comando \verb!\LaTeX!.
```

Il logo "LaTeX" si ottiene con il comando \LaTeX.

```
\begin{verbatim}
Nell'ambiente verbatim
i comandi di \LaTeX,
gli
a capo, gli    spazi,
   i rientri e i
caratteri speciali (\{}%$_&#^~)
non vengono interpretati.
\end{verbatim}
```

```
Nell'ambiente verbatim
i comandi di \LaTeX,
gli
a capo, gli    spazi,
   i rientri e i
caratteri speciali (\{}%$_&#^~)
non vengono interpretati.
```

```
\begin{verbatim*}
Il comando \verb*
e l'ambiente verbatim*
rendono gli spazi così.
\end{verbatim*}
```

```
Il␣comando␣\verb*
e␣l'ambiente␣verbatim*
rendono␣gli␣spazi␣così.
```

Questi strumenti generalmente riescono a soddisfare le esigenze più comuni, ma non permettono alcuna personalizzazione (uso di colore e sfondi particolari per evidenziare la sintassi, di riquadri, possibilità di definire ambienti personali, per esempio). Si rimanda chi ne avesse bisogno ai pacchetti listings (incompatibile con Unicode, però) e fancyvrb.

Per scrivere algoritmi si consigliano i pacchetti algorithm e algpseudocode: il primo, a differenza del secondo, genera oggetti mobili (si veda a questo proposito il paragrafo 6.2 a pagina 86).

4.8 ACRONIMI E GLOSSARI

In lavori particolarmente tecnici e complessi si consiglia di mettere nel *backmatter* un elenco degli acronimi menzionati nel testo e un *glossario*, cioè un elenco alfabetico dei termini specialistici presenti nel documento con le relative definizioni (ed eventualmente un simbolo).

Il pacchetto acronym (se ne veda la documentazione), gestisce efficacemente gli acronimi, generando automaticamente anche i collegamenti ipertestuali tra acronimo nel testo e relativa spiegazione nell'elenco.

Per comporre un vero e proprio glossario esiste il pacchetto glossaries, che tramite il programma MakeIndex genera automaticamente i corrispondenti elenchi.

5 MATEMATICA

Questo capitolo esplora uno dei principali punti di forza di LaTeX, anche se ne intacca solamente la superficie: la composizione di formule matematiche. Si spiegheranno strumenti sufficienti per la *maggior parte* delle esigenze, ma potrebbe darsi che *quella particolare* necessità non trovi risposta in queste pagine. Se così fosse, molto probabilmente la soluzione sta in una delle funzioni del pacchetto amsmath (che non potrà essere descritto per intero, dati i limiti di questo lavoro) o di qualche altro pacchetto dedicato.

La scrittura della matematica e delle altre scienze è regolata da norme che dipendono dalle tradizioni e dalla cerchia dei lettori cui lo scritto è destinato. Queste regole sono raccolte nel mondo dalla normativa ISO (*International Standard Organization*, "Organizzazione Internazionale per lo Standard") e in Italia dalla normativa UNI (Ente Nazionale Italiano di Unificazione). Le norme emesse dall'UNI hanno valore di legge (per cui se un documento per uso legale viene scritto in modo conforme a esse non perde tale valore) ma nella versione originale le si può consultare soltanto tramite un servizio a pagamento.

Di qui in avanti si danno per caricati i pacchetti amsmath e amssymb.

5.1 FORMULE IN LINEA E IN DISPLAY

In linea generale, nelle formule matematiche le variabili vengono stampate in $corsivo\ matematico$, diverso dal *corsivo ordinario* (si noti, però, che il font matematico AMS Euler usato in questa guida è piuttosto simile al tondo, perché volutamente imita la scrittura diritta di un matematico alla lavagna).

Esistono due modi per scrivere la matematica con LaTeX: "in linea" e "in display".

5.1.1 Formule in linea

Una formula "in linea" è un'espressione che LaTeX incorpora nel testo: $\lim_{n \to \infty} \sum_{k=1}^n \frac{1}{k^2} = \frac{\pi^2}{6}$, per esempio. Come si può osservare, il programma fa *il possibile* per comprimerla e mantenere invariata l'interlinea nel capoverso che la contiene. L'esempio seguente mostra come scrivere una formula di questo tipo:

```
Una formula in linea è
un'espressione che \LaTeX{}
incorpora nel testo:
$\lim_{n \to \infty}
\sum_{k=1}^n \frac{1}{k^2} =
\frac{\pi^2}{6}$. Il programma
modifica l'interlinea
del capoverso il meno possibile.
```

Una formula in linea è un'espressione che LaTeX incorpora nel testo: $\lim_{n \to \infty} \sum_{k=1}^n \frac{1}{k^2} = \frac{\pi^2}{6}$. Il programma modifica l'interlinea del capoverso il meno possibile.

Le formule in linea si scrivono tra dollari `$...$` (oppure, ma meno spesso, tra i comandi `\(...\)`). Si consiglia, però, di scrivere in linea soltanto espressioni di piccole dimensioni, come le seguenti:

```
Ci sono voluti secoli per
dimostrare che quando $n>2$
\emph{non} ci sono tre interi
positivi $a$, $b$, $c$ tali
che $a^n+b^n=c^n$.
```

Ci sono voluti secoli per dimostrare che quando $n > 2$ *non* ci sono tre interi positivi a, b, c tali che $a^n + b^n = c^n$.

5.1.2 Formule in display

Una formula "in display", invece, è un'espressione che LaTeX compone su linee a sé, separate dal testo precedente e seguente con spazi bianchi di ampiezza adeguata per "metterla in mostra" e farla risaltare sulla pagina. L'esempio in linea del paragrafo precedente diventa, in display:

$$\lim_{n \to \infty} \sum_{k=1}^{n} \frac{1}{k^2} = \frac{\pi^2}{6}.$$

Come si può osservare, ora la formula è centrata e non compressa, e a numeri e simboli è assegnato lo spazio che devono avere, con un risultato finale di grande respiro.

Le formule di questo tipo vanno scritte tra comandi ancora diversi o in ambienti dedicati alla matematica. Dei molti strumenti esistenti per farlo, oggi *non devono essere più usati*:

- I dollari doppi `$$...$$`, che potrebbero compromettere la corretta spaziatura verticale delle formule o il funzionamento dell'opzione di classe `fleqn` [Fairbairns, 2011].
- Gli ambienti standard `eqnarray` e `eqnarray*` (per sistemi di formule numerate e non numerate rispettivamente), perché tra segno di $=$ e resto del testo inseriscono più spazio del dovuto. È un difetto che gli sviluppatori hanno deciso di conservare in LaTeX 2ε per mantenere la compatibilità con le vecchie versioni del programma.

Il modo corretto di scrivere una formula in display, allora, è quello di racchiuderla in uno dei due ambienti seguenti:

- `equation` per le formule numerate;
- `equation*` (di solito abbreviato in `\[...\]`) per quelle non numerate.

Si considerino gli esempi seguenti:

```
Una formula in display
è un'espressione che \LaTeX{}
compone su linee a sé stanti:
\[
\lim_{n \to \infty}
\sum_{k=1}^n \frac{1}{k^2} =
\frac{\pi^2}{6}.
\]
```

Una formula in display è un'espressione che LaTeX compone su linee a sé stanti:

$$\lim_{n \to \infty} \sum_{k=1}^{n} \frac{1}{k^2} = \frac{\pi^2}{6}.$$

```
Se $f$ è continua e
\[
F(x)=\int_a^x f(t)\,dt,
\]
allora
\begin{equation}
F'(x)=f(x).
\end{equation}
```

Se f è continua e

$$F(x) = \int_a^x f(t)\,dt,$$

allora

$$F'(x) = f(x). \qquad (5.1)$$

Come si può osservare, le formule sono accompagnate da segni di punteggiatura.

Intorno alla punteggiatura nelle formule in display sono fiorite due scuole di pensiero. Alcuni ritengono che non andrebbe *mai* usata, perché superflua e causa di possibili ambiguità nella lettura [Beccari, 2011]. Altri, tra cui chi scrive, ritenendo le formule in genere parte integrante dell'argomentazione, pensano che la punteggiatura aiuti chi legge e che vada quindi messa [Guiggiani e Mori, 2008]. Qualunque delle due scuole si intenda seguire, l'importante è rimanere coerenti con la propria scelta in *tutto* il documento.

Il modo display va riservato a espressioni complesse e di grandi dimensioni (troppo sacrificate tra le righe di un capoverso) e a formule più contenute a cui si voglia dare un risalto particolare.

I comandi `\label` ed `\eqref` permettono di richiamare una formula da altre parti del documento (come avviene per gli oggetti mobili, si veda il paragrafo 6.2 a pagina 86):

```
\begin{equation}
\label{eqn:eulero}
e^{i\pi}+1=0.
\end{equation}
Dalla formula~\eqref{eqn:eulero}
si deduce che\dots
```

$$e^{i\pi}+1=0. \qquad (5.2)$$

Dalla formula (5.2) si deduce che...

5.1.3 Modo matematico e modo testuale

La modalità con cui si scrive la matematica (*modo matematico*) differisce per alcuni aspetti da quella con cui si scrive il testo (*modo testuale*). Ecco i principali.

- Spazi e interruzioni di riga nel testo sorgente non hanno significato, perché LaTeX ne gestisce automaticamente l'inserimento in base alla struttura della formula. Se serve, si possono inserire *a mano* ulteriori spazi con i comandi raccolti nella tabella 21 a pagina 69.

- Nella scrittura delle formule *non sono ammesse righe vuote*.

- LaTeX considera ogni lettera una variabile, e come tale la scrive (cioè in corsivo matematico). Si può comunque inserire un breve testo in tondo e spaziato normalmente in una formula in display scrivendolo nell'argomento del comando `\text` ed esplicitando la spaziatura prima e dopo di esso.

I due esempi seguenti mostrano quanto si è appena descritto:

```
$x+y+z=n$ \\
$ x + y + z = n $
```

$x+y+z=n$
$x+y+z=n$

```
\[
z^2+1=0 \quad
\text{per $z=\pm i$.}
\]
```

$$z^2+1=0 \quad \text{per } z=\pm i.$$

Il comando `\pm` produce $\pm$ (`\mp` produce $\mp$).

5.2 NOZIONI INTRODUTTIVE

Questo paragrafo descrive i comandi più usati per scrivere le formule matematiche. (Per un loro elenco più dettagliato si vedano le tabelle che accompagnano il presente capitolo.)

5.2.1 Raggruppamenti

La maggior parte dei comandi matematici agisce soltanto sul carattere immediatamente successivo. Si evita questo comportamento racchiudendo il testo interessato in un gruppo di parentesi graffe:

```
\[
a^x+y \ne a^{x+y}
\]
```

$$a^x + y \neq a^{x+y}$$

5.2.2 Esponenti, indici e radici

Apici e pedici si scrivono dopo i caratteri ^ e _ rispettivamente:

```
$a_1$ \qquad $x^2$ \qquad
$e^{-\alpha t}$ \qquad
$a^{3}_{ij}$ \\
$e^{x^2} \ne {e^x}^2$
```

a_1 $\qquad$ x^2 $\qquad$ $e^{-\alpha t}$ $\qquad$ a^{3}_{ij}
$e^{x^2} \ne {e^x}^2$

dove il comando `\qquad` produce uno spazio orizzontale di un *quadratone*.

Gli indici di secondo ordine vanno messi in un gruppo di graffe insieme a quelli di ordine superiore: una scrittura come `$x_n_k$` non ha senso.

```
Dalla successione $x_n$
estrarre $x_{n_k}$.
```

Dalla successione x_n estrarre x_{n_k}.

Il simbolo di radice quadrata si ottiene con `\sqrt`, quello di radice n-esima con

```
\sqrt[⟨n⟩]{⟨...⟩}
```

LaTeX calcola automaticamente le dimensioni della radice:

```
$\sqrt{x}$ \qquad
$\sqrt{x^2+\sqrt{y}}$ \qquad
$\sqrt[3]{2}$
```

$\sqrt{x}$ $\qquad$ $\sqrt{x^2+\sqrt{y}}$ $\qquad$ $\sqrt[3]{2}$

5.2.3 Somme, prodotti e frazioni

Il simbolo di sommatoria è generato da `\sum` e quello di produttoria da `\prod`. Gli estremi si scrivono come indici.

```
Trovare il massimo valore
della funzione
\[
f(x_1,\dots,x_n)=
\prod_{k=1}^n x_k
\]
sotto la condizione
\[
\sum_{k=1}^n x_k^2=1.
\]
```

Trovare il massimo valore della funzione

$$f(x_1,\dots,x_n)=\prod_{k=1}^n x_k$$

sotto la condizione

$$\sum_{k=1}^n x_k^2=1.$$

Una frazione, anche complessa, si ottiene semplicemente con il comando

```
\frac{⟨numeratore⟩}{⟨denominatore⟩}
```

Per piccole quantità di "materiale frazionario", a volte la forma n/m è più gradevole sulla pagina.

```
\[
\frac{x^2}{k+1} \qquad
x^{\frac{2}{k+1}} \qquad
x^{1/2}
\]
```

$$\frac{x^2}{k+1} \qquad x^{\frac{2}{k+1}} \qquad x^{1/2}$$

5.2.4 Limiti, derivate e integrali

Il comando

```
\lim_{⟨variabile⟩\to ⟨valore⟩}
```

produce i limiti, e `\infty` produce ∞.

```
\[
\lim_{x\to 0}
\frac{\sin x}{x}=1 \qquad
\lim_{n\to +\infty}f_n=\delta
\]
```

$$\lim_{x\to 0} \frac{\sin x}{x}=1 \qquad \lim_{n\to +\infty}f_n=\delta$$

Le derivate si scrivono con il carattere `'`, che produce un segno di *primo*.

```
\[
y=x^2, \quad y'=2x, \quad y''=2.
\]
```

$$y=x^2, \quad y'=2x, \quad y''=2.$$

Il comando `\int` produce il simbolo di integrale. Gli estremi di integrazione si scrivono come indici, e un indice formato da più di una lettera o una cifra va messo tra parentesi graffe.

```
\[
\int_a^{a+T}f(x)\,dx=
\int_0^T f(x)\,dx
\]
```

$$\int_a^{a+T}f(x)\,dx=\int_0^T f(x)\,dx$$

Come si può osservare, lo spazio sottile `\,` allontana dx da $f(x)$.

Per gli integrali multipli si usano i comandi `\iint`, `\iiint`, `\iiiint` e `\idotsint`.

```
\[
\iint_D f(x,y)\,dx\,dy \qquad
\iiint g \,dx\,dy\,dz
\]
```

$$\iint_D f(x,y)\,dx\,dy \qquad \iiint g \,dx\,dy\,dz$$

Per gli integrali curvilinei lungo curve chiuse si usa `\oint`:

```
\[
\oint f(z)\,dz=2\pi i.
\]
```

$$\oint f(z)\,dz=2\pi i.$$

Tabella 16: Simboli insiemistici (`\bigcup` e `\bigcap` sono operatori).

$\subset$	`\subset`	$\supset$	`\supset`	$\cup$	`\cup`
$\subseteq$	`\subseteq`	$\supseteq$	`\supseteq`	$\cap$	`\cap`
$\in$	`\in`	$\ni$	`\ni` o `\owns`	$\notin$	`\notin`
$\complement$	`\complement`	$\setminus$	`\setminus`	$\emptyset$	`\emptyset`
$\bigcup$	`\bigcup`	$\bigcap$	`\bigcap`		

5.2.5 Insiemi numerici

I simboli degli insiemi numerici si ottengono con `\mathbb` (*blackboard bold*, "nero da lavagna").

```
\[
x\in\mathbb{R}, \quad
z\in\mathbb{C}.
\]
```

$$x \in \mathbb{R}, \quad z \in \mathbb{C}.$$

I simboli usati nell'esempio precedente sono raccolti insieme ad altri simboli insiemistici nella tabella 16.

Se si scrivono nel preambolo le definizioni seguenti (si veda il paragrafo 9.1 a pagina 127)

```
\newcommand{\numberset}{\mathbb}
\newcommand{\N}{\numberset{N}}
\newcommand{\R}{\numberset{R}}
```

per avere $\mathbb{N}$ basta scrivere `\N`, e si può cambiare notazione con un'unica modifica.

5.2.6 Lettere greche

Le lettere greche minuscole e maiuscole si ottengono con i comandi elencati nella tabella 17 a fronte (per l'*omicron* minuscolo si usa il carattere latino *o*, e per le maiuscole che non vi compaiono si usano le corrispondenti maiuscole latine, identiche a quelle greche). Sei di esse, prefissate con `var-`, subiscono una sorte particolare:

- `\varpi` e `\varsigma` non si usano *mai*;
- le altre quattro vanno usate in modo esclusivo: *o* la forma principale *o* la sua variante (ma si ricordi che a differenza di altri Paesi in Europa si usa generalmente la seconda).

È per questo motivo che anche se il font matematico in uso dovesse restituire nello stesso modo le diverse forme (come accade in questa guida) il pericolo di confondersi o di commettere errori non sussisterebbe.

Diventa perciò conveniente ridefinire queste quattro varianti come caratteri normali, scrivendo nel preambolo (si veda il paragrafo 9.1 a pagina 127):

```
\renewcommand{\epsilon}{\varepsilon}
\renewcommand{\theta}{\vartheta}
\renewcommand{\rho}{\varrho}
\renewcommand{\phi}{\varphi}
```

Tabella 17: Lettere greche (il font matematico $\mathcal{AMS}$ Euler usato in questa guida non permette di distinguere tra le diverse forme minuscole di *rho* e *sigma*).

α	`\alpha`	κ	`\kappa`	ς	`\varsigma`
β	`\beta`	λ	`\lambda`	τ	`\tau`
γ	`\gamma`	Λ	`\Lambda`	υ	`\upsilon`
Γ	`\Gamma`	μ	`\mu`	Υ	`\Upsilon`
δ	`\delta`	ν	`\nu`	ϕ	`\phi`
Δ	`\Delta`	ξ	`\xi`	Φ	`\Phi`
ϵ	`\epsilon`	Ξ	`\Xi`	φ	`\varphi`
ε	`\varepsilon`	π	`\pi`	χ	`\chi`
ζ	`\zeta`	Π	`\Pi`	ψ	`\psi`
η	`\eta`	ϖ	`\varpi`	Ψ	`\Psi`
θ	`\theta`	ρ	`\rho`	ω	`\omega`
Θ	`\Theta`	ϱ	`\varrho`	Ω	`\Omega`
ϑ	`\vartheta`	σ	`\sigma`		
ι	`\iota`	Σ	`\Sigma`		

5.2.7 Simboli che sormontano altri simboli

Il comando

```
\overset{⟨primo argomento⟩}{⟨secondo argomento⟩}
```

stampa il simbolo indicato nel ⟨*primo argomento*⟩ rimpicciolito e sovrapposto a quello scritto nel ⟨*secondo argomento*⟩ (di solito un simbolo di relazione binaria), che rimane delle sue dimensioni e nella posizione abituale. Il comando `\underset` fa l'opposto.

```
Il simbolo
\[
\overset{H}{=}
\]
indica l'uguaglianza nel senso
del teorema di de l'H\^{o}pital.
```

Il simbolo

$$\overset{H}{=}$$

indica l'uguaglianza nel senso del teorema di de l'Hôpital.

5.2.8 Barre e accenti

Il comando `\bar` pone un trattino sul suo argomento: il simbolo $\bar{x}$ indica un nome di variabile distinto da x (si veda la tabella 18 nella pagina seguente).

I comandi `\overline` e `\underline` (il secondo dei quali si usa piuttosto raramente) sopralineano e sottolineano rispettivamente tutto il loro argomento: il simbolo $\overline{x}$ indica un operatore applicato alla variabile x:

```
$\bar{x}$ \qquad $\bar{X}$
\qquad $\overline{m+n}$
```

$\bar{x}$ $\qquad$ $\bar{X}$ $\qquad$ $\overline{m+n}$

I comandi `\vec` e `\overrightarrow` agiscono esattamente come `\bar` e `\overline`, ma producono frecce anziché barre orizzontali.

```
$\vec x$ \qquad
$\overrightarrow{AB}$
```

$\vec x$ $\qquad$ $\overrightarrow{AB}$

Tabella 18: Accenti in modo matematico.

$\bar{a}$	`\bar{a}`	$\hat{a}$	`\hat{a}`	$\tilde{a}$	`\tilde{a}`
$\vec{a}$	`\vec{a}`	$\dot{a}$	`\dot{a}`	$\ddot{a}$	`\ddot{a}`
$\check{a}$	`\check{a}`	$\widehat{abc}$	`\widehat{abc}`	$\widetilde{abc}$	`\widetilde{abc}`

Esistono tre tipi di barra verticale, diverse per lo spazio che lasciano tra sé e i simboli circostanti:

- semplice | (ottenibile anche con `\vert`);
- delimitatore sinistro e destro (`\lvert` e `\rvert` rispettivamente);
- relazione binaria (`\mid`) per la divisibilità e il *tale che* negli insiemi.

```
$F(x)|_{x=\gamma(t)}$ \qquad
$\lvert x\rvert$ \\
Se $p\mid n^2$,
allora $p\mid n$.
```

$F(x)|_{x=\gamma(t)}$ $\qquad$ $\lvert x\rvert$
Se $p\mid n^2$, allora $p\mid n$.

I comandi appena esaminati prevedono forme analoghe che stampano barre verticali doppie: `\|` (o `\Vert`), `\lVert`, `\rVert` e `\parallel`.

La barra laterale è prevista talvolta nel calcolo degli integrali:

```
\[
\int_a^b f(x)\,dx = F(x)\big|_a^b
\]
```

$$\int_a^b f(x)\,dx = F(x)\big|_a^b$$

L'altezza della barra va regolata a mano, premettendole uno dei comandi che verranno descritti nel paragrafo 5.4 a pagina 71.

La differenza tra due insiemi si realizza con il comando `\setminus`. Si confrontino le due scritture seguenti:

```
$A\backslash B$ \\
$A\setminus B$
```

$A\backslash B$
$A\setminus B$

Come si può osservare, il primo codice produce una spaziatura (leggermente) sbagliata.

Per il valore assoluto e la norma conviene caricare il pacchetto mathtools e definire nel preambolo due comandi ad hoc:

```
\DeclarePairedDelimiter{\abs}{\lvert}{\rvert}
\DeclarePairedDelimiter{\norma}{\lVert}{\rVert}
```

da usare nel modo seguente:

```
\[
\sum_{n=0}^{+\infty}z^n=
\frac{1}{1-z} \quad
\text{per $\abs{z}<1$.}
\]
```

$$\sum_{n=0}^{+\infty}z^n=\frac{1}{1-z} \quad \text{per $|z|<1$.}$$

```
\[
\norma{x}=
\sqrt{x_1^2+\dots+x_n^2}
\]
```

$$\lVert x\rVert=\sqrt{x_1^2+\dots+x_n^2}$$

Le varianti asterisco dei comandi appena definiti producono delimitatori ad altezza variabile:

Tabella 19: Frecce.

$\leftarrow$	`\leftarrow` o `\gets`	$\longleftarrow$	`\longleftarrow`
$\rightarrow$	`\rightarrow` o `\to`	$\longrightarrow$	`\longrightarrow`
$\leftrightarrow$	`\leftrightarrow`	$\longleftrightarrow$	`\longleftrightarrow`
$\Leftarrow$	`\Leftarrow`	$\Longleftarrow$	`\Longleftarrow`
$\Rightarrow$	`\Rightarrow`	$\Longrightarrow$	`\Longrightarrow`
$\Leftrightarrow$	`\Leftrightarrow`	$\Longleftrightarrow$	`\Longleftrightarrow`
$\mapsto$	`\mapsto`	$\longmapsto$	`\longmapsto`
$\hookleftarrow$	`\hookleftarrow`	$\hookrightarrow$	`\hookrightarrow`
$\uparrow$	`\uparrow`	$\Uparrow$	`\Uparrow`
$\downarrow$	`\downarrow`	$\Downarrow$	`\Downarrow`
$\updownarrow$	`\updownarrow`	$\Updownarrow$	`\Updownarrow`
$\leftleftarrows$	`\leftleftarrows`	$\rightrightarrows$	`\rightrightarrows`
$\nearrow$	`\nearrow`	$\searrow$	`\searrow`
$\swarrow$	`\swarrow`	$\nwarrow$	`\nwarrow`

```
\[
\abs*{\frac{a}{b}} \qquad
\norma*{\frac{u}{\lambda}}
\]
```

$$\left|\frac{a}{b}\right| \qquad \left\|\frac{u}{\lambda}\right\|$$

Esiste un'ulteriore variante per entrambi i comandi:

`\abs[`⟨*grandezza del delimitatore*⟩`]{`⟨*testo*⟩`}`

e

`\norma[`⟨*grandezza del delimitatore*⟩`]{`⟨*testo*⟩`}`

che permette di regolare a mano la grandezza dei delimitatori specificandola nell'argomento facoltativo. Si consideri l'esempio seguente:

```
\[
\abs[\bigg]{\sum_{i=1}^{n}x_i}
\qquad
\abs*{\sum_{i=1}^{n}x_i}
\]
```

$$\biggl|\sum_{i=1}^{n}x_i\biggr| \qquad \left|\sum_{i=1}^{n}x_i\right|$$

Come si può osservare, la prima scrittura presenta delimitatori (leggermente) insufficienti.

Per aggiungere alle variabili un accento matematico, come un cappello o una tilde, si possono usare i comandi della tabella 18 a fronte. I comandi `\widehat` e `\widetilde` stampano rispettivamente simboli di cappello e tilde che sormontano tutto il loro argomento.

5.2.9 Punti, frecce e simboli logici

In matematica esistono due tipi di due punti, diversamente spaziati dal testo circostante:

- semplice : , spaziato come in un'operazione binaria (divisione);
- `\colon`, spaziato come un segno di interpunzione.

Tabella 20: Simboli logici.

$\lor$	`\lor`	$\land$	`\land`	$\neg$	`\neg`
$\exists$	`\exists`	$\nexists$	`\nexists`	$\forall$	`\forall`
$\implies$	`\implies`	$\iff$	`\iff`	$\models$	`\models`

Si confrontino le due scritture seguenti:

```
$f:\R\to\R$ \\
$f\colon\R\to\R$
```

$f : \mathbb{R} \to \mathbb{R}$
$f\colon \mathbb{R} \to \mathbb{R}$

Come si può osservare, il primo codice produce una spaziatura (leggermente) sbagliata.

Per inserire punti ellittici in una formula si usa il comando `\dots`, che li mette automaticamente sulla linea di base del testo o li centra rispetto alla riga a seconda del contesto.

```
\[
x_1,\dots,x_n \qquad
x_1+\dots+x_n
\]
```

$$x_1,\dots,x_n \qquad x_1+\dots+x_n$$

Per ulteriori esempi dell'uso di `\dots`, si rimanda al paragrafo 5.5 a pagina 73.

Oltre alla freccia semplice $\to$, che si ottiene con il comando `\to`, c'è anche quella con il trattino $\mapsto$, che si ottiene con `\mapsto`.

```
$f\colon\R\to\R$ \\
$x\mapsto x^2$
```

$f\colon \mathbb{R} \to \mathbb{R}$
$x \mapsto x^2$

I comandi `\xleftarrow` e `\xrightarrow` stampano frecce che si estendono automaticamente per accordarsi con indici di lunghezza non comune. Entrambi accettano un argomento facoltativo (il pedice) e uno obbligatorio (l'apice) che possono anche rimanere vuoti.

```
\[
\xleftarrow{a} \quad
\xleftarrow[X]{a+b} \quad
\xrightarrow[X+Y+Z]{}
\]
```

$$\xleftarrow{a} \quad \xleftarrow[X]{a+b} \quad \xrightarrow[X+Y+Z]{}$$

La tabella 19 nella pagina precedente raccoglie i principali comandi per ottenere le frecce.

I simboli logici raccolti nella tabella 20 vanno usati *solo* in un lavoro sull'argomento (mentre in tutti gli altri contesti vanno senz'altro preferite le forme estese *se... allora*, *se e solo se*, eccetera).

5.2.10 Spazi in modo matematico

Può accadere, anche se di rado, che la spaziatura scelta da LaTeX per le formule non soddisfi l'autore. Per modificarla si usano i comandi raccolti nella tabella 21 nella pagina successiva. L'unità di misura dello spazio matematico è il *quadrato* (`\quad`), il cui valore standard (si veda la tabella 11 a pagina 40) varia in base al punto dell'espressione matematica in cui viene dato.

Tabella 21: Spazi in modo matematico.

Comando	Tipo di spazio
`\,`	Spazio sottile positivo
`\!`	Spazio sottile negativo
`\quad`	Spazio di un *quadrato*
`\qquad`	Spazio di un *quadratone*

Lo spazio sottile prodotto dal comando `\,` è molto utile in alcune formule. Si confronti

```
\[
\int_a^b f(x) dx, \quad
\sqrt{2} a, \quad
\sqrt{\log x}.
\]
```

$$\int_a^b f(x) dx, \quad \sqrt{2} a, \quad \sqrt{\log x}.$$

con

```
\[
\int_a^b f(x)\,dx, \quad
\sqrt{2}\,a, \quad
\sqrt{\,\log x}.
\]
```

$$\int_a^b f(x)\,dx, \quad \sqrt{2}\,a, \quad \sqrt{\,\log x}.$$

Come si può osservare, il primo codice produce una spaziatura (leggermente) insufficiente.

Il comando `\!` può rivelarsi una finezza utilissima per compensare spaziature spiacevoli in presenza di simboli inclinati (barre di frazione, radicali) preceduti o seguiti da simboli particolari (esponenti, apici o operatori). Si confronti

```
\[
x^2/2, \quad
a/\sin b.
\]
```

$$x^2/2, \quad a/\sin b.$$

con

```
\[
x^2\!/2, \quad
a/\!\sin b.
\]
```

$$x^2\!/2, \quad a/\!\sin b.$$

Come si può osservare, il primo codice produce una spaziatura (leggermente) esagerata.

Si ricorda che interventi come gli ultimi due vanno operati *se e solo se* la spaziatura predefinita non fosse adeguata ai simboli che compaiono nelle formule per migliorare ulteriormente la qualità (già alta) del documento.

5.3 OPERATORI

A differenza delle variabili, stampate in corsivo matematico, LaTeX stampa in tondo le funzioni come sin, cos e log per renderle più visibili sulla pagina (in accordo con le norme ISO-UNI). Esse richiedono una spaziatura particolare (inserita automaticamente dal compositore) che dipende dal tipo di contesto in cui vengono introdotte. In LaTeX, i comandi come `\sin`, `\cos` e `\log` vengono detti *operatori*, e si vedono all'opera negli esempi seguenti:

```
\[
\cos2x=
\frac{1-\sin^2x}{2}
\]
```

$$\cos 2x = \frac{1-\sin^2 x}{2}$$

```
\[
\log\log x, \quad \log(x+y)
\]
```

$$\log\log x, \quad \log(x+y)$$

Si osservi che:

- nella prima formula, fra cos e 2 c'è più spazio che fra 2 e x;
- nella seconda i tre elementi sono separati da uno spazio sottile;
- nella terza non c'è alcuno spazio tra tra log e la parentesi.

Omettendo la barra rovescia nel codice, si otterrebbe "$cos2x$" tutto in corsivo e scorrettamente spaziato, il che in matematica non significa nulla. Soltanto scrivendo gli operatori come si è appena mostrato LaTeX si comporta nel giusto modo e assegna loro font e spazi corretti.

I seguenti sono alcuni operatori predefiniti (la tabella 22 nella pagina successiva ne riporta l'elenco completo):

```
$\arccos x$, $\exp x$,
$\min_{x\in A} f(x)$,
$\det A$, $\log x$, $\tan x$
```

$\arccos x$, $\exp x$, $\min_{x\in A} f(x)$, $\det A$, $\log x$, $\tan x$

I due comandi `\bmod` e `\pmod` riguardano la relazione di congruenza modulo m:

```
$a\bmod b$ \qquad
$a\equiv b \pmod{m}$
```

$a \bmod b$ $\qquad$ $a\equiv b \pmod{m}$

Le pubblicazioni specialistiche introducono continuamente nuove funzioni che devono poter essere definite in qualche modo, non essendo contemplate né da LaTeX né dai pacchetti dedicati. Risolve il problema il comando `\DeclareMathOperator`.

Per esempio, per definire la funzione matematica "sgn" che denoti il segno di un numero reale (funzione non prevista né da LaTeX né da amsmath), si scrive nel preambolo

```
\DeclareMathOperator{\sgn}{sgn}
```

Nel documento, poi, si darà `\sgn` per ottenere "sgn" nel font corretto e adeguatamente spaziato su entrambi i lati.

L'operatore che denota la parte reale di un numero complesso è `\Re`, che produce il simbolo $\Re$. Lo si può ridefinire perché produca il simbolo in tondo anziché in gotico dando nel preambolo i comandi

```
\DeclareMathOperator{\Realpart}{Re}
\renewcommand{\Re}{\Realpart}
```

Se il nuovo operatore presenta indici scritti come nei limiti (lim, sup o max), si usa la variante asterisco di `\DeclareMathOperator`:

```
\DeclareMathOperator*{\argmax}{arg\,max}
```

Tabella 22: Operatori predefiniti.

\min	\max	\inf	\sup	\gcd	\arg
\sin	\cos	\tan	\cot	\sec	\csc
\sinh	\cosh	\tanh	\coth	\exp	\lim
\arcsin	\arccos	\arctan	\log	\lg	\ln
\liminf	\limsup	\deg	\det	\dim	\hom
\ker	\Pr				

5.4 PARENTESI

LaTeX definisce numerosi simboli per parentesi e altri delimitatori. Le parentesi tonde e quadre si scrivono con i corrispondenti caratteri da tastiera, mentre quelle graffe anche in modo matematico devono essere precedute da \. Tutti gli altri delimitatori vengono generati da comandi dedicati.

```
\[
{a,b,c}\ne\{a,b,c\}
\]
```

$$a,b,c \neq \{a,b,c\}$$

Talvolta bisogna aggiustarne a mano le dimensioni: lo si può fare prefissandoli con i comandi `\big`, `\Big`, `\bigg` e `\Bigg`. I comandi `\bigl` (*big left*) e `\bigr` (*big right*) ingrandiscono lievemente le parentesi:

```
\[
\bigl( (x-y)+(x+y) \bigr)
\]
```

$$\bigl((x-y)+(x+y) \bigr)$$

I comandi `\Bigl` e `\Bigr` producono parentesi ancora più grandi:

```
\[
\Bigl(1+\frac{1}{n}\Bigr)^n
\]
```

$$\Bigl(1+\frac{1}{n}\Bigr)^n$$

I comandi `\biggl` e `\biggr` ne generano di più grandi ancora:

```
\[
\biggl(\sum_n x_n^2\biggr)^{1/2}
\]
```

$$\biggl(\sum_n x_n^2\biggr)^{1/2}$$

Se non basta, ci sono anche `\Biggl` e `\Biggr`.

```
$\bigl(\Bigl(\biggl(\Biggl($
\quad
$\bigr\}\Bigr\}\biggr\}\Biggr\}$
\quad
$\big\|\Big\|\bigg\|\Bigg\|$
```

$\bigl(\Bigl(\biggl(\Biggl($ $\quad$ $\bigr\}\Bigr\}\biggr\}\Biggr\}$ $\quad$ $\big\|\Big\|\bigg\|\Bigg\|$

LaTeX può determinare le dimensioni dei delimitatori anche automaticamente, premettendo `\left` a quello di apertura e dando `\right` prima del corrispondente delimitatore di chiusura (nel caso in cui si voglia lasciare l'espressione aperta, si userà `\right.` con il punto finale). Questi comandi funzionano soltanto se usati in coppia e sulla stessa riga. Tuttavia, siccome quasi sempre inseriscono spaziature indesiderate e parentesi più grandi del necessario, si preferisce loro il metodo manuale appena esaminato.

I comandi \overbrace e \underbrace creano delle lunghe graffe orizzontali sopra o sotto un'espressione:

```
\[
\underbrace{1+2+\dots+n}_{{}=
\frac{n(n+1)}{2}}
+(n+1)
\]
```

$$\underbrace{1+2+\dots+n}_{{}=\frac{n(n+1)}{2}}+(n+1)$$

Per scrivere coefficienti binomiali si usa il comando \binom:

```
\[
(a+b)^n=
\sum_{\substack{k\in\N \\
0\le k\le n}}
\binom{n}{k}a^{n-k} b^k
\]
```

$$(a+b)^n=\sum_{\substack{k\in\mathbb{N} \\ 0\le k\le n}}\binom{n}{k}a^{n-k} b^k$$

Il comando

```
\substack{⟨sopra⟩\\⟨sotto⟩}
```

serve per stampare un indice su più righe.

Per i sistemi di equazioni è conveniente definire nel preambolo un ambiente ad hoc (si veda il paragrafo 9.1 a pagina 127):

```
\newenvironment{sistema}%
  {\left\lbrace\begin{array}{@{}l@{}}}%
  {\end{array}\right.}
```

da usare come segue:

```
\[
\begin{sistema}
x+y+z=0 \\
2x-y=1 \\
y-4z=-3
\end{sistema}
\]
```

$$\left\lbrace\begin{array}{@{}l@{}} x+y+z=0 \\ 2x-y=1 \\ y-4z=-3 \end{array}\right.$$

Per gli insiemi è utile il pacchetto braket, che definisce l'apposito comando \Set:

```
\[
\Set{\frac{1}{n^3} | n\in\N}
\]
```

$$\left\{ \frac{1}{n^3} \;\middle|\; n\in\mathbb{N} \right\}$$

e, per le parentesi angolate, \Bra, \Ket e \Braket:

```
\[
\Bra{\psi_n} \quad
\Ket{\psi} \quad
c_n=\Braket{\psi_n | \psi}
\]
```

$$\langle\psi_n| \quad |\psi\rangle \quad c_n=\langle\psi_n|\psi\rangle$$

5.5 VETTORI E MATRICI

I vettori si scrivono di solito in nero tondo (corsivo, secondo le norme ISO-UNI) oppure in semplice corsivo matematico; talvolta, soprattutto nei testi di fisica, sono sormontati da una freccia. Nel primo caso si può usare il comando `\mathbf`; nel secondo il comando `\bm` del pacchetto `bm`; nel terzo il comando `\vec`. Può essere conveniente ridefinire nel preambolo quest'ultimo comando (si veda il paragrafo 9.1 a pagina 127):

```
\renewcommand{\vec}{\bm}
```

In questo modo basta scrivere `\vec{v}` per ottenere $\boldsymbol{v}$ e si può cambiare notazione con un'unica modifica (si veda anche il paragrafo 5.8 a pagina 77).

Le matrici si scrivono negli ambienti `pmatrix`, `bmatrix`, `Bmatrix`, `vmatrix` e `Vmatrix`, che hanno come delimitatori rispettivamente parentesi tonde, quadre (*braces*), graffe (*curly braces*), barre verticali e doppie barre verticali. Esiste anche l'ambiente `matrix` senza delimitatori.

Gli elementi della matrice vengono centrati automaticamente, e righe e colonne si scrivono come una normale tabella `tabular`, ricordando che gli spazi espliciti sono ignorati.

```
\[
\begin{pmatrix}
1 & 2 \\
3 & 4
\end{pmatrix}
\]
```

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$$

```
\[
\begin{bmatrix}
1 & 2 \\
3 & 4
\end{bmatrix}
\]
```

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

```
\[
\begin{vmatrix}
1-x & 2 \\
3   & 4-x
\end{vmatrix}
\]
```

$$\begin{vmatrix} 1-x & 2 \\ 3 & 4-x \end{vmatrix}$$

Si possono scrivere matrici con punti ellittici, come nell'esempio seguente:

```
\[
A=
\begin{bmatrix}
x_{11} & x_{12} & \dots \\
x_{21} & x_{22} & \dots \\
\vdots & \vdots & \ddots
\end{bmatrix}
\]
```

$$A = \begin{bmatrix} x_{11} & x_{12} & \dots \\ x_{21} & x_{22} & \dots \\ \vdots & \vdots & \ddots \end{bmatrix}$$

Come si può osservare, il comando `\vdots` produce tre punti ellittici verticali e `\ddots` tre in diagonale.

Il comando

```
\hdotsfor{⟨n⟩}
```

Tabella 23: Simboli misti (\bigodot, \bigoplus e \bigotimes sono operatori; il font matematico $\mathcal{AMS}$ Euler usato in questa guida rende allo stesso modo \hbar e \hslash).

$\vee$	\vee	$\wedge$	\wedge	$\div$	\div
$\Re$	\Re	$\Im$	\Im	$\aleph$	\aleph
∂	\partial	∇	\nabla	$\cdot$	\cdot
$\hbar$	\hbar	$\hslash$	\hslash	$\circ$	\circ
$\imath$	\imath	$\jmath$	\jmath	$\bullet$	\bullet
ℓ	\ell	$\wp$	\wp	$\surd$	\surd
$\dagger$	\dagger	$\ddagger$	\ddagger	$\ast$	\ast
$\vdash$	\vdash	$\dashv$	\dashv	$\angle$	\angle
$\triangleleft$	\triangleleft	$\triangleright$	\triangleright	$\triangle$	\triangle
$\square$	\square	$\blacksquare$	\blacksquare	$\diamond$	\diamond
$\odot$	\odot	$\oplus$	\oplus	$\otimes$	\otimes
$\bigodot$	\bigodot	$\bigoplus$	\bigoplus	$\bigotimes$	\bigotimes

Tabella 24: Simboli di relazione.

$\le$	\le	$\ge$	\ge	$\sim$	\sim
$\ll$	\ll	$\gg$	\gg	$\simeq$	\simeq
$\prec$	\prec	$\succ$	\succ	$\approx$	\approx
$\preceq$	\preceq	$\succeq$	\succeq	$\asymp$	\asymp
$\parallel$	\parallel	$\perp$	\perp	$\cong$	\cong
$\mid$	\mid	$\propto$	\propto	$\equiv$	\equiv
$\ne$	\ne				

riempie di punti la riga della matrice per $\langle n \rangle$ colonne:

```
\[
\begin{bmatrix}
a_{11} & \dots & a_{1n} \\
a_{21} & \dots & a_{2n} \\
\hdotsfor{3} \\
a_{n1} & \dots & a_{nn}
\end{bmatrix}
\]
```

$$\begin{bmatrix} a_{11} & \dots & a_{1n} \\ a_{21} & \dots & a_{2n} \\ \hdotsfor{3} \\ a_{n1} & \dots & a_{nn} \end{bmatrix}$$

Una piccola matrice in linea si scrive nell'ambiente smallmatrix:

```
Sia $A=\bigl(
\begin{smallmatrix}
a & b \\
c & d
\end{smallmatrix}
\bigr)$
una matrice invertibile.
```

Sia $A=\bigl(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix} \bigr)$ una matrice invertibile.

In questo caso le parentesi vanno aggiunte a mano.

La tabella 23 raccoglie altri simboli matematici di uso comune. Per un loro elenco completo si veda [Pakin, 2009].

5.6 SPEZZARE FORMULE LUNGHE

LaTeX non spezza automaticamente una formula lunga più di una riga. Solo chi l'ha scritta, infatti, ne conosce il ritmo di lettura e sa dov'è più

opportuno andare a capo e se allinearne o meno le varie parti. In generale, tuttavia, vale la regola per cui si può andare a capo:

- dopo i simboli di relazione e, subordinatamente, dopo i simboli di relazione e di operazione binaria (si veda la tabella 24) nelle formule in linea;
- prima dei simboli di relazione e dopo i simboli binari nelle formule in display;
- *mai* dopo gli operatori funzionali (si veda la tabella 22 a pagina 71), i grandi operatori e i delimitatori di apertura.

Per spezzare e raggruppare formule in display, il pacchetto amsmath definisce (fra gli altri) gli ambienti `multline`, `split`, `gather` e `align`, che si descrivono di seguito.

5.6.1 Spezzare formule senza incolonnarle: multline

Per spezzare una formula in più righe senza incolonnarle si usa l'ambiente `multline`.

```
\begin{multline}
f=a+b+c \\
+d+e+g+h \\
+r+s+t.
\end{multline}
```

$$\begin{multline} f=a+b+c \\ +d+e+g+h \\ +r+s+t. \tag{5.3}\end{multline}$$

Si noti che:

- la prima riga viene allineata a sinistra e l'ultima a destra;
- le rimanenti vengono centrate (a meno che non sia attiva l'opzione `fleqn`, che allinea comunque le formule a sinistra rispetto a un margine rientrato);
- il numero progressivo della formula viene stampato nel margine destro in corrispondenza dell'ultima riga.

La variante asterisco `multline*` produce formule dello stesso tipo *non numerate*.

5.6.2 Spezzare formule incolonnandole: split

Per spezzare una formula in più righe incolonnate si usa l'ambiente `split`.

```
\begin{equation}
\begin{split}
a &= b+c-d \\
  &= e-f \\
  &= g+h \\
  &= i.
\end{split}
\end{equation}
```

$$\begin{split} a &= b+c-d \\ &= e-f \\ &= g+h \\ &= i. \end{split} \tag{5.4}$$

Come si può osservare:

- il carattere & incolonna le righe della formula a partire dal punto in cui viene dato (di solito subito prima di un =);
- `split` va *necessariamente* usato dentro un altro ambiente per la matematica in display (come `equation`), responsabile della numerazione della formula (ma si può ottenere una formula di questo tipo anche non numerata).

5.7 RAGGRUPPARE PIÙ FORMULE

Per raggruppare più formule in display, il pacchetto `amsmath` definisce (fra gli altri) gli ambienti `gather` e `align`, descritti di seguito.

5.7.1 Raggruppare formule senza incolonnarle: `gather`

L'ambiente `gather` raggruppa più formule senza incolonnarle, ma centrando e numerando autonomamente ciascuna su una riga a sé e, se necessario, corredandola di un'etichetta.

```
\begin{gather}
a=b+c, \\
V+F-S=2.
\end{gather}
```

$$a = b + c, \tag{5.5}$$
$$V + F - S = 2. \tag{5.6}$$

Si noti che si potrebbe ottenere un risultato "simile" inserendo ciascuna formula del gruppo in un ambiente `equation`, ma la spaziatura tra le righe risulterebbe esagerata. La variante asterisco `gather*` produce formule dello stesso tipo non numerate.

5.7.2 Raggruppare formule incolonnandole: `align`

L'ambiente `align` incolonna gruppi di due o più formule mettendo e numerando ciascuna su una riga a sé, come mostra l'esempio seguente:

```
\begin{align}
a   &= b+c+d, \\
e   &= f, \notag \\
x-1 &= y+z.
\end{align}
```

$$\begin{aligned} a &= b + c + d, && (5.7)\\ e &= f, \\ x - 1 &= y + z. && (5.8) \end{aligned}$$

La variante asterisco `align*` produce formule dello stesso tipo non numerate. (Si noti che lo stesso risultato si ottiene dando `\notag` alla fine della formula interessata).

L'ambiente è utile anche per incolonnare più righe di formule autonome. In tal caso, il carattere & assume due significati diversi a seconda della posizione in cui si trova sulla riga:

- se occupa un posto dispari, indica a LaTeX i punti da incolonnare;
- se occupa un posto pari, è un separatore come in `tabular`.

```
\begin{align}
a &= b,   & c &=d,   & e &=f, \\
u &= v,   & w &=x,   & y &=z.
\end{align}
```

$$\begin{aligned} a &= b, & c &= d, & e &= f, && (5.9)\\ u &= v, & w &= x, & y &= z. && (5.10) \end{aligned}$$

5.7.3 Gli ambienti `gathered` e `aligned`

Per raggruppare più formule con una parentesi, magari da integrare con del testo esplicativo, LaTeX definisce i due ambienti `gathered` e `aligned`, che trattano le espressioni al loro interno come i due ambienti corrispondenti appena esaminati.

```
\[
\left.
\begin{aligned}
a &= b+1 \\
c &= d
\end{aligned}
\right\}
\quad
\text{due equazioni}
\]
```

$$\left.\begin{aligned} a &= b+1 \\ c &= d \end{aligned}\right\}\quad\text{due equazioni}$$

5.7.4 Casi e numerazioni subordinate

L'ambiente `cases` serve per le definizioni fatte per casi. Graffa e allineamento sono automatici; il testo nella seconda colonna va nell'argomento di `\text`.

```
\[
n!=
\begin{cases}
1,       & \text{se $n=0$,} \\
n(n-1)!, & \text{se $n\ge 1$.}
\end{cases}
\]
```

$$n!=\begin{cases} 1, & \text{se $n=0$,} \\ n(n-1)!, & \text{se $n\ge 1$.} \end{cases}$$

L'ambiente `subequations` produce sottonumerazioni:

```
\begin{subequations}
   \label{eqn:schema}
   \begin{align}
   a &= b+c, \\
   c &= d, \label{eqn:sub} \\
   e &= f+g.
   \end{align}
\end{subequations}
Le formule~\eqref{eqn:schema},
e in particolare
la~\eqref{eqn:sub}, \dots
```

$$a = b+c, \qquad (5.11a)$$
$$c = d, \qquad (5.11b)$$
$$e = f+g. \qquad (5.11c)$$

Le formule (5.11), e in particolare la (5.11b), ...

5.8 MODIFICARE STILE E CORPO DEL FONT

In modo matematico, LaTeX armonizza stile e corpo del font con il contesto in cui le formule si trovano. A volte, però, può essere necessario modificare a mano questi due parametri: nei prossimi paragrafi si spiega come farlo.

Modificare lo stile

Gli esempi seguenti mostrano come i comandi per cambiare lo stile del font (raccolti anche nella tabella 25 nella pagina successiva) agiscano su lettere e numeri, ma *non* sui segni di operazione.

Tabella 25: Stili dei font matematici (\mathscr richiede il pacchetto mathrsfs, \mathbb e \mathfrak il pacchetto amsfonts o amssymb). Si noti che alcuni di essi non hanno effetto su minuscole e numeri.

Stile	Codice	Risultato
Tondo	`\mathrm{ABCdef123}`	$\mathrm{ABCdef123}$
Corsivo	`\mathit{ABCdef123}`	$\mathit{ABCdef123}$
Nero	`\mathbf{ABCdef123}`	$\mathbf{ABCdef123}$
Dattilografico	`\mathtt{ABCdef123}`	$\mathtt{ABCdef123}$
Senza grazie	`\mathsf{ABCdef123}`	$\mathsf{ABCdef123}$
Gotico	`\mathfrak{ABCdef123}`	$\mathfrak{ABCdef123}$
Nero da lavagna	`\mathbb{ABC}`	$\mathbb{ABC}$
Calligrafico	`\mathcal{ABC}`	$\mathcal{ABC}$
Manoscritto	`\mathscr{ABC}`	$\mathscr{ABC}$

```
$x+y+2^n M\cos t$ \\
$\mathit{x+y+2^n M\cos t}$ \\
$\mathbf{x+y+2^n M\cos t}$ \\
$\mathrm{x+y+2^n M\cos t}$ \\
$\mathtt{x+y+2^n M\cos t}$ \\
$\mathsf{x+y+2^n M\cos t}$
```

$x+y+2^n M\cos t$
$\mathit{x+y+2^n M\cos t}$
$\mathbf{x+y+2^n M\cos t}$
$\mathrm{x+y+2^n M\cos t}$
$\mathtt{x+y+2^n M\cos t}$
$\mathsf{x+y+2^n M\cos t}$

Spesso i principianti abusano dei simboli in nero, tipograficamente piuttosto “pesanti”. Questi si ottengono con il comando \mathbf, il cui comportamento dipende in larga parte dalla serie di font matematici in uso e di solito ha effetto solo sulle lettere e non sugli altri simboli. Si osservi l’esempio seguente:

```
\[
\mu, M \qquad
\mathbf{\mu}, \mathbf{M}
\]
```

$$\mu, M \qquad —, \mathbf{M}$$

Come si può osservare, il comando non ha effetto sulla lettera μ e restituisce lettere in tondo anziché in corsivo matematico.

Per comporre simboli matematici in nero corsivo si consiglia il comando \bm del pacchetto bm. Si tenga presente che \bm funziona se (e solo se) il font matematico corrente dispone della versione nera di quel simbolo.

```
\[
\mu, M \qquad
\bm{\mu}, \bm{M}
\]
```

$$\mu, M \qquad \boldsymbol{\mu}, \boldsymbol{M}$$

Gli indici letterali vanno scritti in corsivo matematico se rappresentano quantità variabili (cioè se sono dei *simboli*), in tondo se rappresentano apposizioni di una grandezza fisica (cioè se sono semplice *testo*). In quest’ultimo caso si usa il comando \textup (o, in alternativa, il comando \ped definito dall’opzione `italian` di babel, si veda il paragrafo 4.3.2 a pagina 49):

```
\[
V_\textup{eff} \qquad
\psi\ped{incidente}
\]
```

$$V_\mathrm{eff} \qquad \psi_\mathrm{incidente}$$

Si confrontino le due scritture seguenti:

```
$V_{eff}$ \\
$V_\textup{eff}$
```

V_{eff}
$V_\textup{eff}$

Come si può osservare, LaTeX interpreta il pedice nel primo codice come tre variabili da moltiplicare, stampandole in corsivo matematico e spaziandole di conseguenza. Il codice corretto è il secondo.

Modificare il corpo

In modo matematico si può impostare a mano la dimensione del font con le quattro dichiarazioni `\displaystyle`, `\textstyle`, `\scriptstyle` e `\scriptscriptstyle`. Si considerino gli esempi seguenti:

```
\[
\sum_{k=1}^n z^k \qquad
\textstyle\sum_{k=1}^n z^k
\]
$\displaystyle\sum_{k=1}^n z^k$
$\sum_{k=1}^n z^k$
$\scriptstyle\sum_{k=1}^n z^k$
$\scriptscriptstyle
 \sum_{k=1}^n z^k$
```

$$\sum_{k=1}^n z^k \qquad \textstyle\sum_{k=1}^n z^k$$

$\displaystyle\sum_{k=1}^n z^k$ $\sum_{k=1}^n z^k$ $\scriptstyle\sum_{k=1}^n z^k$ $\scriptscriptstyle\sum_{k=1}^n z^k$

```
\[
x_G=
\frac{\displaystyle
      \sum_{i=1}^n m_ix_i}
{\displaystyle\sum_{i=1}^n m_i}
\]
```

$$x_G=\frac{\displaystyle\sum_{i=1}^n m_ix_i}{\displaystyle\sum_{i=1}^n m_i}$$

Come si può osservare, il cambiamento del corpo influisce anche sul modo in cui vengono stampati gli indici.

I due comandi `\dfrac` e `\tfrac` si possono usare come abbreviazioni di `{\displaystyle\frac`⟨...⟩`}` e `{\textstyle\frac`⟨...⟩`}`:

```
$\frac{1}{k}\log_2 c(f)$
\qquad
$\dfrac{1}{k}\log_2 c(f)$
\[
\frac{1}{k}\log_2 c(f) \qquad
\tfrac{1}{k}\log_2 c(f)
\]
```

$\frac{1}{k}\log_2 c(f)$ $\qquad$ $\dfrac{1}{k}\log_2 c(f)$

$$\frac{1}{k}\log_2 c(f) \qquad \tfrac{1}{k}\log_2 c(f)$$

5.9 EVIDENZIARE FORMULE

Per evidenziare una formula con uno sfondo colorato o con una cornice è utile il pacchetto empheq. (Il primo dei due esempi che seguono richiede il pacchetto xcolor.)

```
\newcommand*\mygraybox[1]{%
\colorbox{lightgray}{#1}}
\begin{empheq}[box=%
\mygraybox]{align*}
a &= b \\
E &= mc^2 + \int_a^a x\, dx
\end{empheq}
```

$$\begin{aligned} a &= b \\ E &= mc^2 + \int_a^a x\, dx \end{aligned}$$

```
\begin{empheq}[box=%
\fbox]{align*}
a &= b \\
E &= mc^2 + \int_a^a x\, dx
\end{empheq}
```

$$\boxed{\begin{aligned} a &= b \\ E &= mc^2 + \int_a^a x\, dx \end{aligned}}$$

5.10 DIAGRAMMI COMMUTATIVI

I diagrammi commutativi sono particolari oggetti a metà strada fra un insieme di espressioni matematiche e un disegno che le mette in relazione tra di loro.

Il pacchetto amscd definisce un ambiente `CD` (da usare soltanto all'interno di un ulteriore ambiente per formule in display) con cui si possono disegnare diagrammi commutativi piani e senza frecce diagonali. Per diagrammi più complessi ci sono pacchetti più sofisticati, come xypdf.

```
\[
\begin{CD}
A       @>f>> B \\
@V{g}VV       @VV{h}V \\
C       @>>k> D \\
\end{CD}
\]
```

5.11 ENUNCIATI E DIMOSTRAZIONI

Di qui in avanti si dà per caricato *anche* il pacchetto amsthm.

Nella scrittura della matematica è utile poter disporre di un metodo per introdurre e numerare definizioni, teoremi e strutture simili. I tipi di enunciato non sono predefiniti, ma vanno dichiarati dall'autore, chiamato a prendere alcune decisioni globali:

- stabilire il tipo di enunciato da inserire (per esempio, definizioni e teoremi);
- assegnare un nome a ogni ambiente (per esempio, `definizione` e `teorema`; non si può usare `def`, perché è già un comando di base di LaTeX);
- titolare gli enunciati (per esempio, con *Definizione* e *Teorema*).

Il comando `\newtheorem`, dato *nel preambolo*, permette di fare le relative dichiarazioni globali e prevede due forme di definizione:

\newtheorem{⟨*nome dell'enunciato*⟩}{⟨*titolo*⟩}[⟨*sezione*⟩]

oppure, in alternativa,

\newtheorem{⟨*nome dell'enunciato*⟩}[⟨*numerato come*⟩]{⟨*titolo*⟩}

dove:

- ⟨*nome dell'enunciato*⟩ è una parola chiave per identificare con facilità l'enunciato anche successivamente;
- ⟨*titolo*⟩ specifica il titolo dell'enunciato che verrà stampato nel documento;

- ⟨*sezione*⟩ specifica il livello di sezionamento (`chapter` o `section`, di regola) a cui collegare la numerazione dell'enunciato;

- in ⟨*numerato come*⟩ si scrive il nome di un enunciato dichiarato in precedenza, in modo che quello nuovo ne prosegua la numerazione.

La variante asterisco `\newtheorem*` produce enunciati non numerati.

Il testo dell'enunciato va messo entro i consueti comandi di apertura e chiusura d'ambiente, e un'eventuale specificazione si scrive nell'argomento facoltativo immediatamente *dopo* il comando di apertura (verrà stampato fra parentesi tonde), così:

```
\begin{⟨nome dell'enunciato⟩}[⟨eventuale specificazione⟩]
...
\end{⟨nome dell'enunciato⟩}
```

Il pacchetto amsthm prevede tre stili predefiniti per gli enunciati (`plain`, `definition` e `remark`) i cui dettagli tipografici dipendono dalla classe di documento in uso, anche se in linea di massima il primo dei tre stampa il proprio contenuto in corsivo, mentre gli altri due lo lasciano in tondo. Di seguito si riportano le principali categorie di enunciato, ciascuna associata al suo stile più tipico:

`plain` Per teoremi, lemmi, corollari, proposizioni, congetture, criteri, leggi e algoritmi.

`definition` Per definizioni, condizioni, problemi ed esempi.

`remark` Per osservazioni e annotazioni.

Si possono stampare gli enunciati con stili diversi dal predefinito `plain` dividendoli in gruppi per tipo di enunciato e premettendo a ciascun gruppo il comando `\theoremstyle`. (Si può personalizzare lo stile dell'enunciato con il comando `\newtheoremstyle`.)

A questo punto le nozioni teoriche dovrebbero bastare. Alcuni esempi mostreranno quanto si è appena esaminato. Scrivendo nel preambolo

```
\theoremstyle{definition}
\newtheorem{definizione}{Definizione}
```

e

```
\theoremstyle{plain}
\newtheorem{teorema}{Teorema}
```

gli ambienti `definizione` e `teorema` si usano così:

```
\begin{definizione}[di Gauss]
Si dice \emph{matematico} colui
per il quale è ovvio che
$\int_{-\infty}^{+\infty}
e^{-x^2}\,dx=\sqrt{\pi}$.
\end{definizione}
\begin{teorema}
I matematici, se ce ne sono,
sono molto rari.
\end{teorema}
```

Definizione 1 (di Gauss)**.** Si dice *matematico* colui per il quale è ovvio che $\int_{-\infty}^{+\infty} e^{-x^2}\,dx=\sqrt{\pi}$.

Teorema 1. *I matematici, se ce ne sono, sono molto rari.*

```
Il seguente teorema è a tutti
ben noto.
\begin{teorema}[di Pitagora]
La somma dei quadrati costruiti
sui cateti è uguale
al quadrato costruito
sull'ipotenusa.
\end{teorema}
La dimostrazione è lasciata
per esercizio.
```

Il seguente teorema è a tutti ben noto.

Teorema 2 (di Pitagora)**.** *La somma dei quadrati costruiti sui cateti è uguale al quadrato costruito sull'ipotenusa.*

La dimostrazione è lasciata per esercizio.

Come si può osservare:

- LaTeX numera automaticamente in nero ogni enunciato e ne conclude l'intestazione con un punto fermo;
- ogni enunciato è staccato dal resto del testo senza rientro;
- lo stile delle definizioni è in tondo, quello dei teoremi in corsivo.

L'esempio seguente riguarda la numerazione di tre enunciati consecutivi, il primo e il terzo dei quali sono dello stesso tipo, ma il secondo no. Scrivendo nel preambolo

```
\theoremstyle{plain}
\newtheorem{legge}{Legge}
\newtheorem{decreto}[legge]{Decreto}
```

gli ambienti `legge` e `decreto` si usano come segue:

```
\begin{legge}
\label{lex:capo}
Il capo ha ragione.
\end{legge}
\begin{decreto}[Aggiornamento
alla legge~\ref{lex:capo}]
Il capo ha \emph{sempre} ragione.
\end{decreto}
\begin{legge}
Se il capo ha torto, vedere la
legge~\ref{lex:capo}.
\end{legge}
```

Legge 1. *Il capo ha ragione.*

Decreto 2 (Aggiornamento alla legge 1)**.** *Il capo ha* sempre *ragione.*

Legge 3. *Se il capo ha torto, vedere la legge 1.*

Si osservi che il numero assegnato a *Decreto* prosegue la numerazione di *Legge* anziché cominciarne una nuova, perché l'argomento facoltativo `[legge]` nella definizione del teorema assegna entrambi gli enunciati allo stesso contatore. Come di consueto, ci si può riferire anche a un enunciato da un punto qualsiasi del documento semplicemente assegnandogli un'etichetta.

Se si desidera introdurre un enunciato *Murphy*, per esempio, la cui numerazione sia collegata al paragrafo corrente, basta specificare nell'argomento facoltativo l'opzione `section`, in questo modo:

```
\newtheorem{murphy}{Murphy}[section]
```

L'ambiente `murphy` così definito si usa come di consueto:

```
\begin{murphy}
Se esistono due o più modi per
fare una cosa, e uno di questi
può creare una catastrofe,
allora qualcuno lo sceglierà.
\end{murphy}
```

Murphy 5.11.1. *Se esistono due o più modi per fare una cosa, e uno di questi può creare una catastrofe, allora qualcuno lo sceglierà.*

Per scrivere una dimostrazione si usa l'ambiente `proof`, che ne segnala la fine stampando un quadratino. Si osservi l'esempio seguente:

```
\begin{teorema}[Sorpresa]
Si ha che $\log(-1)=0$.
\end{teorema}
\begin{proof}
Si ha che $\log(-1)^2=2\log(-1)$.
Ma è anche vero che
$\log(-1)^2=\log(1)=0$.
Perciò $2\log(-1)=0$, da cui la
tesi.
\end{proof}
```

Teorema 3 (Sorpresa)**.** *Si ha che* $\log(-1) = 0$.

Dimostrazione. Si ha che $\log(-1)^2 = 2\log(-1)$. Ma è anche vero che $\log(-1)^2 = \log(1) = 0$. Perciò $2\log(-1) = 0$, da cui la tesi. □

Per sostituire la scritta *Dimostrazione* con un'altra, per esempio *Soluzione*, basta scrivere

```
\begin{proof}[Soluzione]
```

Il comando `\qedhere` colloca correttamente il simbolo di fine dimostrazione anche se questa termina con una formula in display. Si confrontino i due esempi seguenti:

```
\begin{proof}
Basta usare la formula
\[
E=mc^2.
\]
\end{proof}
```

Dimostrazione. Basta usare la formula

$$E = mc^2.$$

□

```
\begin{proof}
Basta usare la formula
\[
E=mc^2.\qedhere
\]
\end{proof}
```

Dimostrazione. Basta usare la formula

$$E = mc^2. \qquad \square$$

Come si può osservare, il primo dei due presenta il quadratino collocato in modo (leggermente) sbagliato: la scrittura corretta è la seconda.

5.12 ALTRE SCIENZE

Le unità di misura del Sistema Internazionale si inseriscono con i comandi del pacchetto siunitx (se ne veda la documentazione), che permette di regolarne molto finemente il formato e di cambiare il risultato nel documento finito operando un'unica modifica nel preambolo anziché agire a mano su ciascuna unità di misura.

Dato che le convenzioni tipografiche italiane prevedono la virgola e non il punto (predefinito dal pacchetto) come separatore decimale, il pacchetto va caricato così:

```
\usepackage[output-decimal-marker={,}]{siunitx}
```

Ecco qualche esempio:

```
\SI{23.4}{kg.m.s^{-2}} \\
$r=\SI{0.8768(11)e-15}{m}$ \\
\si{\joule\per\mole\per\kelvin}\\
\si{j.mol^{-1}.K^{-1}}
```

$23{,}4\,\mathrm{kg\,m\,s^{-2}}$
$r = 0{,}8768(11) \times 10^{-15}\,\mathrm{m}$
$\mathrm{J\,mol^{-1}\,K^{-1}}$
$\mathrm{j\,mol^{-1}\,K^{-1}}$

```
\SI{100}{\celsius} \\
\ang{1;2;3}
```

100 °C
1°2′3″

Tutti i precedenti comandi funzionano sia in modo testuale sia in modo matematico. Si tenga presente, tuttavia, che se le unità di misura da inserire nel documento sono poche si possono sempre scrivere senza usare i comandi di siunitx, come si può vedere nell'esempio seguente:

```
L'auto viaggiava a 65\,km/h.
```

L'auto viaggiava a 65 km/h.

Se si prevede di inserire i simboli dei gradi Celsius o di quelli sessagesimali, si consiglia di caricare *anche* il pacchetto textcomp, che abilita la ricerca di ° nel documento finito.

Per comporre semplici formule chimiche è utile il pacchetto mhchem.

```
\ce{H2O} \qquad
\ce{^{227}_{90}Th+} \qquad
\ce{C6H5-CHO} \\[1ex]
\ce{SO4^2- + Ba^2+ -> BaSO4 v}
```

H_2O $\qquad$ ${}^{227}_{90}Th^{+}$ $\qquad$ $C_6H_5{-}CHO$

$SO_4^{2-} + Ba^{2+} \longrightarrow BaSO_4\downarrow$

Il pacchetto chemfig compone strutture molecolari, formule di reazione e tutta la chimica di cui si può aver bisogno.

```
\setcrambond{2pt}{}{}
\setatomsep{2em}
\chemname{\chemfig{*6((-HO)-=%
-(-(<[::60]OH)-[::-60]-%
[::-60,,,2] HN-[::+60]CH_3)=%
-(-HO)=)}}{Adrenalina}
```


Adrenalina

Il pacchetto feynMF disegna i grafici di Feynman.

```
\begin{fmffile}{esempio}
\begin{fmfgraph*}(40,25)
  \fmfleft{em,ep}
  \fmf{fermion}{em,Zee,ep}
  \fmf{photon}{Zee,Zff}
  \fmf{fermion}{fb,Zff,f}
  \fmfright{fb,f}
  \fmfdot{Zee,Zff}
\end{fmfgraph*}
\end{fmffile}
```

Chi si occupa di informatica ha bisogno delle parentesi angolate $\langle\ldots\rangle$, che si ottengono con i comandi `\langle` e `\rangle` in modo matematico.

Per la scrittura di tutte le altre discipline scientifiche si rimanda al catalogo tematico dei pacchetti disponibile su ☞ SAROVAR.

6 TABELLE E FIGURE

Tabelle e figure sono tra gli "oggetti" più usati nei documenti e tra i più problematici, perché di regola non si possono spezzare su più pagine. In questo capitolo, che spiega come servirsene senza sorprese, si danno per caricati i seguenti pacchetti: booktabs e caption per le tabelle; graphicx per le figure.

6.1 STRUMENTI FONDAMENTALI

Per inserire tabelle e figure in un documento da comporre con LaTeX esistono tre strumenti essenzialmente:

- l'ambiente standard `tabular`, per tabelle che contengono prevalentemente testo;
- l'ambiente standard `array`, per tabelle che contengono prevalentemente matematica;
- il comando `\includegraphics` del pacchetto graphicx, per includere nel documento le figure quando sono file *esterni* a esso (come tutte quelle di questa guida).

Li si vede all'opera nei tre esempi seguenti, i cui comandi e ambienti verranno spiegati nei prossimi paragrafi.

```
La tabella
\begin{center}
\begin{tabular}{ll}
\toprule
Alcaloide & Origine \\
\midrule
atropina  & belladonna \\
morfina   & papavero \\
nicotina  & tabacco \\
\bottomrule
\end{tabular}
\end{center}
mostra l'origine vegetale
di alcuni alcaloidi.
```

La tabella

Alcaloide	Origine
atropina	belladonna
morfina	papavero
nicotina	tabacco

mostra l'origine vegetale di alcuni alcaloidi.

```
La tabella
\[\begin{array}{ll}
\toprule
f(x)   & f'(x) \\
\midrule
x^n    & nx^{n-1} \\
e^x    & e^x \\
\sin x & \cos x \\
\bottomrule
\end{array}\]
mostra le derivate
di alcune funzioni elementari.
```

La tabella

$f(x)$	$f'(x)$
x^n	nx^{n-1}
e^x	e^x
$\sin x$	$\cos x$

mostra le derivate di alcune funzioni elementari.

```
La figura
\begin{center}
\includegraphics[width=%
0.5\textwidth]{Rettili}
\end{center}
riproduce
l'incisione su legno
\emph{Tassellazione
del piano con rettili}
di M.~Escher.
```

La figura

riproduce l'incisione su legno *Tassellazione del piano con rettili* di M. Escher.

Si noti che:

- Tutti e tre *non* cominciano un nuovo capoverso, ma producono un'unità tipografica indivisibile che il programma tratta come se fosse un unico carattere, e che nel caso di `tabular` e `\includegraphics` va centrata rispetto alla giustezza del testo mettendola nell'ambiente `center`;
- Una tabella `array` va racchiusa a sua volta tra comandi matematici: se in testo, di solito si usano `\[...\]`, che sostituiscono l'ambiente `center` (ma se l'opzione di classe `fleqn` è attiva, la tabella non risulterà più centrata); se mobile, invece, si usano `$...$`.
- Quando richiesta, si consiglia di assegnare agli oggetti una larghezza *relativa* esprimendola con una frazione della giustezza della riga di testo normale stabilita dalla classe in uso (`\textwidth`, ma si veda anche il paragrafo 9.3.4 a pagina 135). Le unità di misura esplicite, infatti, la "congelano", con gli ovvi inconvenienti quando si dovesse cambiare classe di documento o aumentare le colonne di composizione.
- `center` si omette anche quando si vuole in linea una figura particolarmente piccola, come mostra l'esempio seguente:

```
Il simbolo della mela
\includegraphics[width=%
0.10\textwidth]{apple}
è il logo di Apple.
```

Il simbolo della mela è il logo di Apple.

6.2 OGGETTI IN TESTO E FUORI TESTO

In tipografia esistono due tipi di oggetto: "in testo" e "fuori testo".

6.2.1 Tabelle e figure in testo

Osservando con attenzione gli esempi del paragrafo precedente, si possono notare le caratteristiche degli oggetti "in testo", i quali:

- appartengono al flusso del discorso e non possono esserne scorporati senza comprometterne la comprensione;
- non prevedono la didascalia (proprio perché la loro funzione è spiegata dal contesto) né riferimenti espliciti a sé stessi da altre parti del documento;
- devono essere, perciò, quanto mai chiari e intuitivi.

Apparentemente innocui, oggetti di questo tipo possono comportare in realtà problemi di impaginazione a volte irrisolvibili. Si immagini, per esempio, di essere arrivati quasi alla fine della pagina, e di dover inserire "lì", perché il discorso lo richiede, una figura alta cinque centimetri avendone però soltanto tre a disposizione: va da sé che lì la figura *non* ci può stare in nessun modo. Se lo spazio fisico non c'è, non non lo si può inventare! Questa situazione, si badi bene, non si verifica solo con LaTeX, ma si dà indipendentemente dal programma in uso per scrivere. Come fare?

Una prima soluzione ingenua (e da non attuare) potrebbe essere quella di cominciare una nuova pagina ogni volta che si incappa in un oggetto che non sta in quella corrente; in questo modo, però, le pagine interrotte rimarrebbero parzialmente bianche, con un risultato tipografico insoddisfacente.

Ecco perché gli oggetti in testo devono essere *eccezionali* (tornano utili per mettere un logo "proprio lì" e in pochissime altre circostanze) e di piccole o piccolissime dimensioni.

6.2.2 Tabelle e figure fuori testo

Si risolve il problema rendendo gli oggetti "fuori testo" (o "mobili"; in inglese *floating*, "galleggianti"), cioè affidandone la gestione a LaTeX, che li metterà nel punto *per lui* migliore rispettando rigorosamente l'ordine in cui oggetti dello stesso tipo sono definiti nel testo sorgente: sulla pagina corrente se ci stanno, oppure in pagine successive a quella in cui finirebbero, riempiendo lo spazio che rimane con l'altro materiale a disposizione. Questa modalità è molto vantaggiosa, perché garantisce l'ottimale riempimento della pagina tipico di LaTeX.

A differenza di quelli in testo, gli oggetti fuori testo:

- non appartengono al flusso del discorso e per esigenze tipografiche possono essere spostati altrove dal punto esatto in cui stanno nel testo sorgente;

- prevedono obbligatoriamente una didascalia che ne illustri il contenuto e un numero progressivo al quale potersi riferire direttamente da altre parti del documento.

Indipendentemente dalle dimensioni del documento e degli oggetti, perciò, si raccomanda di includerli *sempre* così (come si è fatto in tutta questa guida), vincendo quanto prima le iniziali e comprensibili perplessità derivanti dal vederli molto spesso in un punto diverso da quello in cui li si è definiti, come si spiegherà tra poco.

Nei prossimi paragrafi si spiegano gli ambienti mobili e come comportarsi durante la stesura del documento; nel paragrafo 10.2 a pagina 144, invece, si mostra che cosa (eventualmente) si può fare durante la sua revisione per migliorarne ulteriormente la qualità tipografica e risolvere collocazioni poco gradite.

Gli ambienti `table` *e* `figure`

Per rendere mobile un oggetto basta inserirne il relativo codice nell'ambiente standard `table`

```
\begin{table}[⟨preferenze di collocazione⟩]
...
\end{table}
```

se è una tabella, oppure in quello `figure`

```
\begin{figure}[⟨preferenze di collocazione⟩]
...
\end{figure}
```

se è una figura. Come si può notare, entrambi accettano un argomento facoltativo ⟨*preferenze di collocazione*⟩, il cui funzionamento verrà spiegato nel paragrafo 6.2.2 nella pagina successiva. (Se si sta componendo un documento a due colonne si possono usare le varianti asterisco dei due ambienti, che mettono l'oggetto sull'intera pagina e non sulla singola colonna di composizione.) I due ambienti appena esaminati permettono di usare alcuni comandi importanti, descritti di seguito.

Il comando

```
\caption{⟨didascalia⟩}
```

stampa l'intestazione *Tabella* o *Figura* e la ⟨*didascalia*⟩ dell'oggetto, assegnandogli anche un numero progressivo.

I due comandi

```
\listoftables
\listoffigures
```

producono *nel punto in cui li si dà* rispettivamente la sezione contenente l'elenco delle tabelle o delle figure (esattamente come fa `\tableofcontents` per l'indice generale del documento) con la relativa testatina. Le loro voci sono le stesse didascalie degli oggetti o, se troppo lunghe, una loro versione ridotta da scrivere nell'argomento facoltativo di `\caption`:

```
\caption[⟨didascalia breve⟩]{⟨didascalia normale⟩}
```

Si badi bene a inserire i due indici appena visti solo se il numero degli oggetti nel documento è rilevante o è importante per il lettore poterli ritrovare agevolmente: un elenco di sole due figure, per esempio, non avrebbe alcuna utilità. Anche per ottenere questi due indici nel documento finito servono *due* composizioni successive.

Il comando `\label`, da dare sempre *dopo* il corrispondente `\caption`, assegna all'oggetto un'etichetta per i riferimenti incrociati (si veda il paragrafo 3.10 a pagina 37).

Codici tipo

Il modo migliore per introdurre un oggetto mobile nel codice sorgente è scriverne il relativo ambiente tra capoversi, cioè preceduto e seguito da una riga vuota. Ecco un esempio tipico per una tabella `tabular` (in modo del tutto analogo si compone una tabella `array`, ricordandosi di racchiuderla tra dollari) con il relativo richiamo:

```
\dots qui finisce un capoverso.

\begin{table}
\caption{⟨...⟩}
\label{tab:esempio}
\centering
\begin{tabular}{⟨...⟩}
...
\end{tabular}
\end{table}

La tabella~\ref{tab:esempio} è un esempio di tabella mobile.
```

E uno per una figura:

```
\dots qui finisce un capoverso.

\begin{figure}
\centering
\includegraphics[width=0.5\textwidth]{...}
\caption{⟨...⟩}
\label{fig:esempio}
\end{figure}

La figura~\ref{fig:esempio} è un esempio di figura mobile.
```

Come si può osservare:

- per centrare un oggetto mobile sulla pagina si usa `\centering`, perché l'ambiente `center` lascia tra testo e oggetto uno spazio verticale eccessivo (ma adeguato per un oggetto in testo);

- la corretta posizione della didascalia varia a seconda dell'oggetto cui è apposta: la tradizione italiana la vuole prima di una tabella e dopo una figura, e in queste posizioni *deve* essere messa anche nel testo sorgente.

Le classi standard prevedono la didascalia sempre *sotto* l'oggetto, perciò LaTeX non la "riconosce" se messa sopra e la restituisce in modo non appropriato stampandogliela a ridosso. Risolve il problema il pacchetto caption, da impostare come segue:

```
\usepackage{caption}
\captionsetup{tableposition=top,figureposition=bottom,font=small}
```

dove:

- `tableposition=top` ordina al programma di inserire uno spazio adeguato tra didascalia e tabella;

- `figureposition=bottom` è l'opzione analoga alla precedente per le figure;

- `font=small` produce didascalie in corpo più piccolo, secondo l'uso italiano;

- quando le opzioni sono numerose o la loro forma lo richiede, si possono scrivere con il comando `\captionsetup`.

Si noti che molte classi non standard modificano il rapporto tra ambiente mobile e didascalia per questioni di stile. Il risultato finale, perciò, *potrebbe* non corrispondere a quello appena descritto: si faccia qualche prova.

Che cosa fare durante la stesura

Durante la stesura del documento, si sa, bisogna concentrarsi sul contenuto del lavoro, lasciando a LaTeX il compito di occuparsi di tutto il resto. La gestione degli oggetti non si sottrae a questa regola, per cui in prima battuta si consiglia di inserirli tutti *senza specificare alcuna preferenza di collocazione* (si veda poco sotto): il risultato è generalmente ottimo. In linea di massima si troveranno gli oggetti "abbastanza" vicini al punto in cui stanno nel testo sorgente, e in particolare:

- quasi sempre in cima alla pagina;

- raramente in basso;

Tabella 26: Preferenze di collocazione per gli oggetti mobili.

Preferenza	Chiede a LaTeX di mettere l'oggetto
`h`	Qui (h*ere*), se possibile
`t`	In cima (t*op*) alla pagina
`b`	In fondo (b*ottom*) alla pagina
`p`	In una pagina di soli oggetti mobili (p*age of floats*)
`!`	Il più possibile dove vorrebbe l'autore

- in un pagina di soli oggetti mobili se sono grandi (secondo i parametri di LaTeX).

Basta sfogliare un qualunque libro ben composto per verificare che le cose stanno proprio così.

Non ci si lamenti per non vederli esattamente dove li si è definiti! Non è un difetto di LaTeX, questo, ma un vantaggio, proprio come lo è il non doversi preoccupare di numerare a mano sezioni e pagine o il non dover pensare a quanto spazio ci va tra un titolo e il testo successivo.

I risultati automatici sono *quasi sempre* migliori di quelli che si potrebbero ottenere cercando di collocare gli oggetti a mano. Quando i gusti dell'autore non coincidono con quelli del programma, tuttavia, si possono usare le ⟨*preferenze di collocazione*⟩ raccolte nella tabella 26, che "suggeriscono" a LaTeX come si vorrebbero vedere gli oggetti sulle pagine del proprio documento.

LaTeX seguirà i suggerimenti nell'ordine in cui li trova (fa eccezione la preferenza `[p]`, come si spiega nel paragrafo 10.2 a pagina 144). Di seguito si propongono un paio tra le possibilità più usate:

- `[tp]` se non si vuole nessun oggetto in fondo alla pagina;
- `[htp]` se si vuole che LaTeX cerchi come prima cosa di mettere l'oggetto esattamente lì dove lo si è inserito (se è sufficientemente piccolo, di solito si viene accontentati).

Ecco ora le opzioni da evitare *sempre*:

- `[h]` o, peggio, `[!h]` *possono* funzionare solo con oggetti molto piccoli; in caso contrario, l'oggetto viene stampato alla fine del capitolo (o del documento) portandosi dietro tutti gli altri inseriti successivamente (si tenga *ben* presente questo comportamento);
- `[t]` e `[b]` da sole, perché è buona regola dare al programma almeno un paio di possibilità (ma si ammette la sola `[p]`).

6.3 TABELLE

Comporre tabelle di altissima qualità è una delle specialità di LaTeX, ma i suoi comandi standard sono piuttosto limitati. Numerosi pacchetti, però, ne definiscono di nuovi e più avanzati con cui si può personalizzare finemente il proprio lavoro. Questo paragrafo, basato su [Mori, 2006], cui si rimanda per ogni approfondimento, spiega come usare gli uni e gli altri e affronta gli aspetti principali dell'argomento.

6.3.1 Indicazioni generali

Regole generali di composizione

Le regole seguenti permettono di ottenere una tabella ben composta:

Tabella 27: Tabella che viola le regole generali.

D	P	u	β	G
.500 m	269.8 kg	.000674 m	1.79	.04089 Pa
1.50 m	421.0 kg	.001035 m	3.59	″
10.0 m	640.2 kg	.001565 m	7.18	″

Tabella 28: Tabella che rispetta le regole generali.

D (m)	P (kg)	u (m)	β	G (Pa)
0,500	269,8	0,000 674	1,79	0,040 89
1,50	421,0	0,001 035	3,59	0,040 89
10,0	640,2	0,001 565	7,18	0,040 89

- non usare *mai* filetti verticali;
- evitare filetti doppi;
- scrivere *sempre* le unità di misura nell'intestazione della colonna (e non nel corpo della tabella);
- non usare *mai* le virgolette per ripetere il contenuto di una cella;
- far precedere *sempre* il separatore decimale da una cifra.

A queste si possono aggiungere le seguenti:

- incolonnare i numeri al separatore decimale, se presente, che nei documenti in italiano è la virgola;
- usare i numeri maiuscoli nelle tabelle numeriche (si veda il paragrafo A.4.2 a pagina 153).

Così vuole la tradizione tipografica, in contrasto con la cattiva abitudine, purtroppo oggi molto diffusa, di comporre le tabelle come se fossero parti di un foglio elettronico. Per capire quanto sia importante rispettare queste regole, si confrontino le tabelle 27 e 28.

Separare le celle e chiudere le righe

Le celle di una tabella vanno separate tra loro con il carattere separatore & e le righe *devono* terminare con il comando `\\`, pena l'errore. Si noti che se una riga ha meno celle piene di quante sono le colonne, può essere chiusa dopo l'ultima cella utile.

Filetti orizzontali: `booktabs`

Per stampare i filetti orizzontali della tabella il pacchetto `booktabs` definisce tre nuovi comandi che sostituscono lo standard `\hline`, dalla resa tipografica insoddisfacente per via dello spazio troppo risicato che risulta tra filetti e testo nelle celle. Questi comandi non vogliono `\\` dopo di sé, stampano filetti di spessore differente (le righe prodotte da `\midrule`, infatti, sono più sottili delle altre) e vanno dati secondo un ordine rigoroso:

1. `\toprule` stampa il primo filetto della tabella;
2. `\midrule` stampa il filetto interno (o, ripetendolo, i filetti, ma non se ne abusi);
3. `\bottomrule` stampa l'ultimo filetto.

Tabella 29: Descrittori standard delle colonne.

Descrittore	Spiegazione
`l`	Allinea il contenuto della cella a sinistra (*left*)
`c`	Centra il contenuto della cella (*center*)
`r`	Allinea il contenuto della cella a destra (*right*)
`p`	Giustifica un testo lungo entro una ⟨*larghezza*⟩
`*`	Ripetitore di descrittori

Codice ordinato

Anche se LaTeX non richiede di incolonnare le celle nel testo sorgente, si consiglia di farlo ugualmente: un codice ordinato facilita eventuali modifiche successive e diminuisce la probabilità di commettere errori.

6.3.2 Tabelle standard: `tabular` e `array`

In linea generale, una tabella che contiene prevalentemente testo va composta dentro l'ambiente `tabular`; una tabella che contiene prevalentemente matematica va composta dentro l'ambiente `array`. Entrambi si comportano in modo molto simile, come si può osservare negli esempi seguenti:

```
\begin{tabular}{lcr}
\toprule
Grandezza & Simbolo & Unità \\
\midrule
forza     & $F$     & newton \\
energia   & $E$     & joule \\
tensione  & $V$     & volt \\
\bottomrule
\end{tabular}
```

Grandezza	Simbolo	Unità
forza	F	newton
energia	E	joule
tensione	V	volt

```
\[
\begin{array}{cc}
\toprule
f(x)   & \text{Una primitiva} \\
\midrule
e^x    & e^x \\
\cos x & \sin x \\
\sin x & -\cos x \\
\bottomrule
\end{array}
\]
```

$f(x)$	Una primitiva
e^x	e^x
$\cos x$	$\sin x$
$\sin x$	$-\cos x$

Alle osservazioni del paragrafo 6.1 a pagina 85 si aggiungano le seguenti:

- `tabular` e `array` richiedono un argomento, detto *preambolo della tabella*, formato da un certo numero di *caratteri descrittori*, ciascuno dei quali definisce il comportamento di un *tipo di colonna* come spiegato nella tabella 29;
- in `tabular`, eventuali formule matematiche si scrivono con i comandi per le formule in linea, per esempio fra dollari `$...$`;
- in `array`, un eventuale testo si scrive nell'argomento del comando `\text` del pacchetto amsmath, che va dunque caricato.

Si noti che per "alleggerire" il codice, nel preambolo di una tabella si può *sempre* usare la scrittura

Tabella 30: Tabella con colonna p.

Forza	Una forza è una grandezza fisica che si manifesta nell'interazione di due o più corpi materiali, che cambia lo stato di quiete o di moto dei corpi stessi.
Momento polare	Il momento polare di una forza rispetto a una determinata origine è definito come il prodotto vettoriale tra il vettore posizione (rispetto alla stessa origine) e la forza.

`*{`⟨*n*⟩`}{`⟨*descrittori*⟩`}`

equivalente a ⟨*n*⟩ colonne consecutive del tipo indicato dai ⟨*descrittori*⟩.

6.3.3 Celle con testo troppo lungo

Le tabelle migliori si ottengono lasciando loro la propria larghezza naturale. È ciò che fanno i tre descrittori `l c r`, allargando automaticamente la cella in base al suo contenuto. Se quest'ultimo è costituito da un testo troppo lungo, però, la tabella eccede la giustezza della riga e LaTeX lo notifica con il relativo avviso. Per queste celle, *e soltanto per queste*, non si possono più usare le colonne appena viste, ma bisogna ricorrere ad altri strumenti:

- a un descrittore `p{`⟨*larghezza*⟩`}`, che permette di stabilire a priori la larghezza *di una sola colonna*;
- al pacchetto tabularx, che permette di stabilire a priori la larghezza *dell'intera tabella*.

Colonne di larghezza prefissata

Il codice seguente, che mostra all'opera il descrittore `p{`⟨*larghezza*⟩`}`, produce la tabella 30:

```
\begin{tabular}{lp{0.5\textwidth}}
\toprule
\textbf{Forza} & Una forza è una grandezza fisica che si manifesta
nell'interazione di due o più corpi materiali, che cambia lo stato
di quiete o di moto dei corpi stessi. \\
\midrule
\textbf{Momento polare} & Il momento polare di una forza rispetto a
una determinata origine è definito come il prodotto vettoriale tra
il vettore posizione (rispetto alla stessa origine) e la forza. \\
\bottomrule
\end{tabular}
```

Si noti che:

- `\textbf` stampa il proprio argomento in nero;
- per impostazione predefinita, il contenuto di una colonna `p` viene giustificato e sillabato automaticamente e l'intestazione, se presente, viene allineata a sinistra per impostazione predefinita;
- se presenti, le colonne `l`, `c` e `r` rimangono della loro larghezza naturale.

Tabella 31: Tabella di larghezza prefissata ottenuta con tabularx.

Forza	Una forza è una grandezza fisica che si manifesta nell'interazione di due o più corpi materiali, che cambia lo stato di quiete o di moto dei corpi stessi.
Momento polare	Il momento polare di una forza rispetto a una determinata origine è definito come il prodotto vettoriale tra il vettore posizione (rispetto alla stessa origine) e la forza.

Tabella 32: Tabella con due colonne della della stessa larghezza ottenuta con tabularx.

Periodo	Fenomeni geologici	Biosfera
Giurassico	Periodo caratterizzato da variazioni del livello del mare; prevalenza delle terre emerse in America, Asia, Australia.	Fauna: compaiono i primi marsupiali; dominano i grandi rettili (dinosauri). Flora: predominano le conifere.
Triassico	Intensa l'erosione dei continenti; profonde fratture da cui escono lave che originano altopiani estesi.	Fauna: si diffondono i rettili; nei mari prosperano pesci e invertebrati. Flora: si sviluppano alghe caratteristiche.

Tabelle di larghezza prefissata

Il pacchetto tabularx, che carica il pacchetto array (si veda il paragrafo 6.3.8 a pagina 99), definisce l'omonimo ambiente `tabularx` e un nuovo tipo di colonna `X`, che alle caratteristiche delle colonne `p` esaminate nel paragrafo precedente aggiunge un vantaggio: è LaTeX a calcolarne automaticamente la larghezza sulla base della larghezza complessiva assegnata *all'intera tabella*. Infatti tabularx richiede *obbligatoriamente* un secondo argomento in cui indicare tale larghezza (nei prossimi esempi pari a `\textwidth`, ma sono ammessi anche altri valori, da impostare come già spiegato nel paragrafo 6.1 a pagina 85). La tabella 31, identica alla 30 nel contenuto, è utile per osservare all'opera il pacchetto. Di seguito se ne propone il codice.

```
\begin{tabularx}{\textwidth}{lX}
\toprule
\textbf{Forza} & Una forza è una grandezza fisica che si manifesta
nell’interazione di due o più corpi materiali, che cambia lo stato
di quiete o di moto dei corpi stessi. \\
\midrule
\textbf{Momento polare} & Il momento polare di una forza rispetto a
una determinata origine è definito come il prodotto vettoriale tra
il vettore posizione (rispetto alla stessa origine) e la forza. \\
\bottomrule
\end{tabularx}
```

La tabella 32, prodotta con il codice seguente, mostra come a *tutte* le colonne `X`, se più d'una, LaTeX assegni la stessa larghezza *indipendentemente* dalle altre colonne presenti.

```
\begin{tabularx}{\textwidth}{lXX}
\toprule
Periodo & Fenomeni geologici & Biosfera \\
\midrule
\textbf{Giurassico} & Periodo caratterizzato da variazioni del
livello del mare; prevalenza delle terre emerse in America, Asia,
```

Tabella 33: Tabella con colonna S.

Espressione	Valore
π	3,1416
π^{π}	36,46
$\pi^{\pi^{\pi}}$	80 662,7

```
Australia. & Fauna: compaiono i primi marsupiali; dominano i grandi
rettili (dinosauri). Flora: predominano le conifere. \\
\midrule
\textbf{Triassico} & Intensa l'erosione dei continenti; profonde
fratture da cui escono lave che originano altopiani estesi.
& Fauna: si diffondono i rettili; nei mari prosperano pesci e
invertebrati. Flora: si sviluppano alghe caratteristiche. \\
\bottomrule
\end{tabularx}
```

6.3.4 Colonne di soli numeri: siunitx

Il pacchetto siunitx gestisce in modo molto potente e flessibile anche la resa tipografica dei numeri nelle tabelle, definendo un nuovo tipo di colonna S *specifica per dati numerici* che si comporta come segue:

- nei numeri di cinque o più cifre separa automaticamente le cifre a gruppi di tre per migliorarne la leggibilità (si veda il paragrafo A.4.2 a pagina 153);
- per impostazione predefinita, mette il separatore decimale al centro della colonna, di cui espande flessibilmente entrambi i margini in base alla lunghezza del numero;
- spazia correttamente il separatore decimale (se c'è) tra parte intera e decimale del numero (si veda il paragrafo A.4.2 a pagina 153).

Lo si vede all'opera nella tabella 33, ottenuta con il seguente codice:

```
\usepackage{siunitx}
\sisetup{output-decimal-marker={,}}

\begin{tabular}{cS}
\toprule
Espressione      & {Valore} \\
\midrule
$\pi$            & 3.1416 \\
$\pi^{\pi}$      & 36.46 \\
$\pi^{\pi^{\pi}}$ & 80662.7 \\
\bottomrule
\end{tabular}
```

Si noti che:

- si possono scrivere le opzioni di siunitx anche nell'argomento del comando `\sisetup` dato immediatamente dopo aver caricato il pacchetto (alternativa di solito preferita quando sono particolarmente numerose);
- un eventuale testo in una colonna S (di solito nell'intestazione) si scrive tra parentesi graffe per non "confondere" il pacchetto, che lì si aspetterebbe dei numeri.

Tra le opzioni, numerose, per il trattamento dei numeri, se ne segnala qui una, utile per risolvere alcune situazioni in cui il comportamento predefinito delle colonne S risulta inadeguato. Nell'esempio seguente si nota uno spazio tra le colonne decisamente esagerato:

```
\begin{tabular}{SS}
\toprule
{$P$}  & {$u$} \\
{(kg)} & {(m)} \\
\midrule
269.8  & 0.000674 \\
421.0  & 0.001035 \\
640.2  & 0.001565 \\
\bottomrule
\end{tabular}
```

P (kg)	u (m)
269,8	0,000 674
421,0	0,001 035
640,2	0,001 565

Risolve il problema l'opzione `table-format=`⟨*valore*⟩, da assegnare *a ciascuna* colonna S nel preambolo della tabella. La sintassi completa dell'opzione è:

`table-format=`⟨*prima*⟩`.`⟨*dopo*⟩

dove:

- ⟨*prima*⟩ è il numero di cifre intere del numero che nella colonna ha la parte intera più lunga;
- '`.`' è il separatore decimale;
- ⟨*dopo*⟩ è il numero di cifre decimali del numero che nella colonna ha la parte decimale più lunga.

La tabella precedente diventa:

```
\begin{tabular}%
   {S[table-format=3.1]%
    S[table-format=1.6]}
\toprule
{$P$}  & {$u$} \\
{(kg)} & {(m)} \\
\midrule
269.8  & 0.000674 \\
421.0  & 0.001035 \\
640.2  & 0.001565 \\
\bottomrule
\end{tabular}
```

P (kg)	u (m)
269,8	0,000 674
421,0	0,001 035
640,2	0,001 565

Nello stesso modo è stata prodotta la tabella 28 a pagina 91.

Se in un documento le tabelle numeriche sono poche o pochissime, siunitx potrebbe rivelarsi troppo laborioso: si abbia cura allora di allineare i numeri a destra e di assegnare loro lo stesso numero di cifre decimali.

6.3.5 Celle multicolonna

Il comando

`\multicolumn{`⟨*n*⟩`}{`⟨*descrittore*⟩`}{`⟨*testo*⟩`}`

sostituisce a ⟨*n*⟩ celle successive un'unica cella, il cui ⟨*testo*⟩ viene stampato nei modi specificati con il ⟨*descrittore*⟩. Lo si vede all'opera nel prossimo esempio, che produce la tabella 34 a fronte:

Tabella 34: Tabella con cella multicolonna.

Nome		Carica
Particella	Antiparticella	(e)
elettrone	positrone	∓ 1
protone	antiprotone	± 1
neutrone	antineutrone	0

Tabella 35: Tabella con celle multiriga.

Famiglia	Particella	Simbolo
leptoni	elettrone	e
	muone	μ
	tau	τ
	neutrino	ν
barioni	protone	p
	neutrone	n

```
\begin{tabular}{llc}
\toprule
\multicolumn{2}{c}{Nome}    & Carica \\
Particella & Antiparticella & (e) \\
\midrule
elettrone  & positrone      & $\mp 1$ \\
protone    & antiprotone    & $\pm 1$ \\
neutrone   & antineutrone   & $0$ \\
\bottomrule
\end{tabular}
```

Si può usare il comando, utile soprattutto nell'intestazione della tabella, per modificare il formato anche di *una sola* cella: in questo caso, a ⟨*n*⟩ si sostituisce 1.

6.3.6 Celle multiriga

Il comando

```
\multirow{⟨n⟩}*{⟨testo⟩}
```

(che richiede il pacchetto omonimo) crea una cella alta ⟨*n*⟩ righe, il cui ⟨*testo*⟩ verrà centrato verticalmente. Il codice

```
\begin{tabular}{clc}
\toprule
Famiglia                & Particella & Simbolo \\
\midrule
\multirow{4}*{leptoni} & elettrone  & $e$  \\
                        & muone      & $\mu$ \\
                        & tau        & $\tau$ \\
                        & neutrino   & $\nu$ \\
\midrule
\multirow{2}*{barioni} & protone    & $p$ \\
                        & neutrone   & $n$ \\
\bottomrule
\end{tabular}
```

Tabella 36: Tabella con celle multiriga e multicolonna.

Elemento	Strati		
	K	L	M
idrogeno	1		
litio	2	1	
sodio	2	8	1

produce la tabella 35 nella pagina precedente. (Si noti, però, che il testo in una cella multiriga *non sempre* viene perfettamente centrato rispetto alle righe circostanti.)

I due comandi si possono combinare. Il codice

```
\begin{tabular}{lccc}
\toprule
\multirow{2}*{Elemento} & \multicolumn{3}{c}{Strati} \\
\cmidrule(lr){2-4}
                        & K   & L   & M \\
\midrule
idrogeno                & $1$ &     & \\
litio                   & $2$ & $1$ & \\
sodio                   & $2$ & $8$ & $1$ \\
\bottomrule
\end{tabular}
```

produce la tabella 36.

Il comando

```
\cmidrule{⟨n⟩-⟨m⟩}
```

disegna un filetto orizzontale dalla sinistra della colonna ⟨*n*⟩-esima fino alla destra della colonna ⟨*m*⟩-esima. Per migliorare leggermente la resa della tabella, si può specificare subito dopo `\cmidrule` un argomento facoltativo ⟨*troncamento*⟩ *fra parentesi tonde*:

```
\cmidrule(⟨troncamento⟩){⟨n⟩-⟨m⟩}
```

che accetta tre possibilità: `r`, `l` o `rl`. Si sta dicendo a LaTeX che il filetto va "rasato" a destra (`r`), a sinistra (`l`) o a entrambe le estremità (`rl`).

6.3.7 Spaziare a mano righe e colonne

Quando (molto di rado) i risultati di LaTeX non soddisfano completamente, si può migliorare la resa tipografica della tabella con piccoli aggiustamenti manuali. I casi, di solito, sono due:

1. le righe della tabella appaiono troppo ravvicinate;
2. le colonne della tabella appaiono troppo ravvicinate.

Risolve il primo caso il comando

```
\\[⟨altezza⟩]
```

che, sostituito allo standard `\\` a fine riga *nel corpo della tabella*, abbassa la riga immediatamente successiva (e solo quella) di uno spazio verticale pari ad ⟨*altezza*⟩. Di solito si inserisce questo spazio supplementare per rimediare a lievi sovrapposizioni (specialmente di formule matematiche in display) oppure per far "respirare" una tabella come si è fatto nella tabella 39 a

pagina 103, dove si sono allontanati leggermente i blocchi di testo altrimenti troppo vicini.

Si osservi come la spaziatura automatica nella tabella seguente è troppo risicata

```
\[
\begin{array}{cc}
\toprule
f(x)      & f'(x) \\
\midrule
\log x    & \dfrac{1}{x} \\
\arctan x & \dfrac{1}{1+x^2} \\
\bottomrule
\end{array}
\]
```

$f(x)$	$f'(x)$
$\log x$	$\dfrac{1}{x}$
$\arctan x$	$\dfrac{1}{1+x^2}$

e come la si può correggere molto semplicemente con

```
\[
\begin{array}{cc}
\toprule
f(x)      & f'(x) \\
\midrule
\log x    & \dfrac{1}{x} \\[2ex]
\arctan x & \dfrac{1}{1+x^2} \\
\bottomrule
\end{array}
\]
```

$f(x)$	$f'(x)$
$\log x$	$\dfrac{1}{x}$
$\arctan x$	$\dfrac{1}{1+x^2}$

Il comando \dfrac richiede il pacchetto amsmath.

Non accade praticamente *mai* di dover aumentare la distanza tra due colonne. Se proprio ce ne fosse bisogno, si può usare il separatore

```
@{⟨larghezza⟩}
```

che, dato *nel preambolo della tabella* tra i descrittori delle due colonne che si desidera allontanare, inserisce uno spazio orizzontale pari a ⟨*larghezza*⟩. Si noti che *in questo caso* la ⟨*larghezza*⟩ va indicata con modalità leggermente diverse dal solito: con le consuete espressioni (valore e unità di misura) scritte nell'argomento di \hspace{⟨...⟩} oppure con comandi di spaziatura "fissa" (\quad e \qquad, per esempio).

Le due istruzioni appena esaminate si possono eseguire insieme e anche più di una volta nella stessa tabella, là dove serve.

6.3.8 Personalizzare le colonne: array

I diversi tipi di colonna esaminati fino a qui (gli standard l, c, r e p e gli "speciali" X e S) riescono a soddisfare la maggior parte delle esigenze, ma non si possono personalizzare. Si immagini, per esempio, di dover scrivere il contenuto di un'intera colonna in nero: è sicuramente più vantaggioso, soprattutto se le celle sono numerose, definire questo stile una volta per tutte anziché ripeterne il comando in ognuna.

Risolve il problema il *pacchetto* array (da non confondere con l'omonimo ambiente standard), che per personalizzare le colonne definisce le due istruzioni >{⟨*dichiarazione iniziale*⟩} e <{⟨*dichiarazione finale*⟩}, la cui sintassi completa è

```
>{⟨dichiarazione iniziale⟩}⟨descrittore⟩<{⟨dichiarazione finale⟩}
```

Tabella 37: Tabella ottenuta con array.

$\int \cos x \, dx$	$\sin x + c$
$\int e^x \, dx$	$e^x + c$
$\int \sec^2 x \, dx$	$\tan x + c$

dove:

- >{⟨*dichiarazione iniziale*⟩} va data prima del ⟨*descrittore*⟩ per dire a LaTeX di eseguirla *prima* del contenuto della colonna;
- ⟨*descrittore*⟩ è un descrittore di colonna, di quelli già esaminati o eventualmente definito con \newcolumntype (si veda più sotto);
- >{⟨*dichiarazione finale*⟩} va data dopo il ⟨*descrittore*⟩ per dire a LaTeX di eseguirla *dopo* il contenuto della colonna.

Le istruzioni appena viste si possono usare nel preambolo *della tabella* se servono ogni tanto; se invece un documento richiede numerose tabelle con colonne personalizzate, è più conveniente definirle una volta per tutte nel preambolo *del documento* con il comando \newcolumntype, la cui sintassi completa è

```
\newcolumntype{⟨carattere⟩}{>{⟨dichiarazione⟩}⟨descrittore⟩<{⟨dichiarazione⟩}}
```

dove ⟨*carattere*⟩ è una lettera, non già in uso nel documento come descrittore di colonna, che indica il nuovo descrittore da richiamare nel preambolo della tabella nel modo consueto.

Nei prossimi paragrafi si vedono le dichiarazioni all'opera.

Modificare lo stile del font di una colonna

Le caratteristiche di array tornano utili per modificare lo stile del font di una colonna: basta sostituire a ⟨*dichiarazione iniziale*⟩ una delle dichiarazioni elencate nella tabella 43 a pagina 131. Il codice

```
\begin{tabular}{>{\bfseries}lp{0.5\textwidth}}
\toprule
Forza & Una forza è una grandezza fisica che si manifesta
nell'interazione di due o più corpi materiali, che cambia lo stato
di quiete o di moto dei corpi stessi. \\
\midrule
Momento polare & Il momento polare di una forza rispetto a una
determinata origine è definito come il prodotto vettoriale tra il
vettore posizione (rispetto alla stessa origine) e la forza. \\
\bottomrule
\end{tabular}
```

produce la tabella 30 a pagina 93, ottenuta in precedenza specificando a mano lo stile della prima colonna all'inizio di ogni riga.

Colonne di sola matematica

I comandi seguenti

```
\newcolumntype{L}{>{$}l<{$}}
\newcolumntype{C}{>{$}c<{$}}
\newcolumntype{R}{>{$}r<{$}}
```

definiscono tre nuove colonne `L`, `C` e `R` nelle quali le formule matematiche vengono rispettivamente allineate a sinistra, centrate e allineate a destra. Se si preferisce il formato in display, basta aggiungere il relativo comando alla definizione della nuova colonna. Il codice

```
\newcolumntype{L}{>{$\displaystyle}l<{$}}
\newcolumntype{C}{>{$}c<{$}}

\begin{tabular}{LC}
\toprule
\int\cos x\,dx     & \sin x + c \\
\midrule
\int e^x dx        & e^x + c \\
\midrule
\int \sec^2 x \,dx & \tan x + c \\
\bottomrule
\end{tabular}
```

produce la tabella 37 a fronte.

Per ulteriori dettagli su array si consiglia la lettura della sua documentazione e di [Gregorio, 2010].

6.3.9 Tabelle con note

Il comando `\footnote`, che in un testo stampa le note al piede, non funziona nell'ambiente `tabular` e, come per la maggior parte delle cose in LaTeX, questa limitazione ha ottime ragioni per esistere. Come si sa, una tabella dovrebbe essere *sempre* inserita fuori testo in un documento: una nota al piede la vincolerebbe invece alla pagina in cui si trova.

Il luogo più adatto per mettere eventuali annotazioni è la didascalia della tabella, ma a volte le tabelle con note esplicative sono imposte dal relatore o dall'editore. A seconda dei due casi in cui ci si trova di solito:

1. se si vuole che la nota alla tabella segua la numerazione delle altre note presenti nel testo si può usare il pacchetto footnote;

2. se invece si vuole una numerazione autonoma, per esempio con lettere al posto dei numeri, si può usare il pacchetto ctable.

6.3.10 Tabelle grandi

Se le dimensioni della tabella finita eccedono quelle della gabbia del testo in lunghezza, in larghezza o in entrambe, si prospettano soluzioni diverse a seconda della dimensione in eccesso.

Se la tabella è *troppo lunga*, si può:

- ridurre il corpo del font;
- spezzarla su più pagine.

Se la tabella è *troppo larga*, si può:

- ridurre il corpo del font;
- ruotarla;
- spezzarla su più pagine.

A ogni caso si possono applicare più soluzioni contemporaneamente.

Tabella 38: Tabella con font di corpo inferiore (`footnotesize`) al resto del testo (`normalsize`).

Forza	Una forza è una grandezza fisica che si manifesta nell'interazione di due o più corpi materiali, che cambia lo stato di quiete o di moto dei corpi stessi.
Momento polare	Il momento polare di una forza rispetto a una determinata origine è definito come il prodotto vettoriale tra il vettore posizione (rispetto alla stessa origine) e la forza.

Ridurre il corpo del font

Per ridurre il corpo del font in una tabella (ma non nell'eventuale didascalia) si usano le stesse dichiarazioni elencate nella tabella 44 a pagina 132, con l'avvertenza di darle:

- *fuori* da `tabular` se la tabella è *in testo* (subito dopo la tabella una nuova dichiarazione ripristinerà il font corrente);
- *dentro* `table` se la tabella è *mobile* (si veda il paragrafo 6.2 a pagina 86).

Il codice

```
\begin{table}[tb]\footnotesize
\caption{⟨...⟩}\label{⟨...⟩}\centering
\begin{tabular}{lp{0.5\textwidth}}
...
\end{tabular}
\end{table}
```

produce la tabella 38. (Se ne confronti il risultato con la tabella 30 a pagina 93.)

Ruotare una tabella

Per ruotare di 90° una tabella è utile il pacchetto rotating, che definisce l'ambiente `sidewaystable` da usare nel modo seguente:

```
\begin{sidewaystable}
\caption{⟨...⟩}\label{⟨...⟩}\centering
\begin{tabular}
...
\end{tabular}
\end{sidewaystable}
```

Si noti che una tabella di questo tipo è mobile e occupa *sempre* una pagina a sé, come mostra la tabella 9 a pagina 32. Per ruotare le immagini il pacchetto definisce l'analogo ambiente `sidewaysfigure`.

Tabelle su più pagine

Una tabella *deve* sempre poter stare in una sola pagina: se è più lunga, LaTeX taglia le parti in eccesso. Il pacchetto longtable può risolvere il problema e ripartire una tabella su più pagine, anche se oggetti di questo tipo, si ricordi, andrebbero usati *soltanto se inevitabili*.

A differenza di altri pacchetti simili (come supertabular e xtab) di solito longtable riesce a usare su ogni pagina la stessa larghezza di riga, corrispondente a quella della riga più lunga della tabella: il risultato finale, però, potrebbe richiedere alcune composizioni successive. Si noti che anche se

può avere una didascalia, una tabella longtable *non* è mobile: se ne valutino sempre con attenzione opportunità e posizione, dunque, e nel caso in cui vada a pagina nuova molto prima della fine di quella corrente si sia pronti a riformulare il testo nei dintorni per bilanciare le due pagine.

La tabella 39, che mostra il pacchetto all'opera, riporta tutte le parti di una longtable nell'ordine in cui vanno scritte, ma va ricordato che non sono obbligatorie né si deve abusarne: la didascalia non sempre è necessaria, e anche intestazioni e piedi sono facoltativi, ma si consiglia di scrivere almeno uno dei due per segnalare al lettore che la stessa tabella occupa più di una pagina.

Tabella 39: Esempio di tabella ripartita su più pagine.

Comando	Effetto
`\begin{longtable}`	Comincia la tabella. L'ambiente si comporta come `tabular`, ma in più, dopo aver composto ogni riga controlla l'altezza complessiva della tabella: se supera quella della pagina, vengono inseriti automaticamente il contenuto del piede (*foot*) e il comando `\end{tabular}`, e la tabella continua su una nuova pagina con l'intestazione scelta (*head*).
`[`⟨*carattere*⟩`]`	Va eventualmente specificato tra l'apertura dell'ambiente e il preambolo della tabella per impostarne la posizione sulla pagina (`c` se al centro, `l` se a sinistra e `r` se a destra). Non indicandolo, la tabella viene centrata per impostazione predefinita.
`{`⟨*preambolo*⟩`}`	È il consueto preambolo nel quale indicare i descrittori già esaminati nei paragrafi precedenti.
`\caption{`⟨*didascalia*⟩`} \\`	Se specificato, assegna alla tabella una didascalia, anche nella versione `[`⟨*didascalia breve*⟩`]` (si veda il paragrafo 6.2.2 a pagina 87).
`\label{`⟨*etichetta*⟩`} \\`	Assegna alla tabella un'etichetta, utile per i riferimenti incrociati (si veda lo stesso paragrafo).
⟨*intestazione iniziale*⟩ `\\` `\endfirsthead`	Specifica l'⟨*intestazione iniziale*⟩, cioè l'intestazione della tabella nella prima pagina in cui compare.
⟨*intestazione normale*⟩ `\\` `\endhead`	Specifica l'⟨*intestazione normale*⟩, cioè l'intestazione della tabella dalla seconda pagina in poi (*Continua dalla pagina precedente*, per esempio).
⟨*piede normale*⟩ `\\` `\endfoot`	Specifica il ⟨*piede normale*⟩, cioè il testo che deve comparire alla fine di ogni pagina (*Continua nella prossima pagina*).

Continua nella prossima pagina

Continua dalla pagina precedente

Comando	Effetto
⟨*piede finale*⟩ \\ \endlastfoot	Specifica il ⟨*piede finale*⟩, cioè il testo che deve apparire subito dopo l'ultima riga della tabella (*Si conclude dalla pagina precedente*).
⟨*corpo della tabella*⟩	Qui si mette il contenuto della tabella, separando le colonne e terminando le righe nei modi consueti.
\end{longtable}	Termina la tabella.

Si conclude dalla pagina precedente

Il codice seguente ha prodotto la tabella 39 nella pagina precedente:

```
\begin{longtable}{lp{0.48\textwidth}}
% intestazione iniziale
\caption{Esempio di tabella ripartita su più pagine.}
\label{tab:longtable} \\
\toprule
Comando & Effetto \\
\midrule
\endfirsthead
% intestazione normale
\multicolumn{2}{l}{\footnotesize\itshape
Continua dalla pagina precedente} \\
\toprule
Comando & Effetto \\
\midrule
\endhead
% piede normale
\midrule
\multicolumn{2}{r}{\footnotesize\itshape
Continua nella prossima pagina} \\
\endfoot
% piede finale
\bottomrule
\multicolumn{2}{r}{\footnotesize\itshape
Si conclude dalla pagina precedente} \\
\endlastfoot
% corpo della tabella
...  & ... \\
...
...  & ... \\
\end{longtable}
```

Per evitare errori nella composizione, si ricordi di terminare con \\ le righe di intestazioni, piedi e didascalia (o etichetta, se presente). Si noti, infine, che in casi particolarissimi anche una `longtable` può essere ruotata inserendola nell'ambiente `landscape` del pacchetto pdflscape (se ne veda la documentazione), ma come di consueto si raccomanda di non abusare di questa possibilità.

Per ulteriori approfondimenti si consiglia la lettura di [Gregorio, 2010].

6.4 FIGURE

Le figure rientrano tra gli argomenti più studiati dalle guide a LaTeX, tanto che ne esistono di specifiche, cui si rimanda per ogni approfondimento.

Questi oggetti presentano almeno due problemi diversi, riguardanti:

1. il *tipo di file* da introdurre nel documento (verrà trattato in questo paragrafo);

2. la *collocazione* della figura sulla pagina (verrà trattato nel paragrafo successivo).

Di qui in avanti si dà per caricato il pacchetto graphicx.

6.4.1 Immagini vettoriali e bitmap

Si possono dividere le figure in due grandi classi: le immagini *vettoriali* e le immagini *bitmap* [Mori, 2007].

Immagini vettoriali

Le immagini vettoriali sono descritte da forme, possono essere scalate e deformate senza perdere in definizione e sono adatte soprattutto per grafici e schemi, argomento non considerato in questa guida per via della sua complessità (gli strumenti più diffusi per produrre grafici e schemi con LaTeX sono l'ambiente standard `picture`, i pacchetti PGF, TikZ, PSTricks, Xy-pic e i programmi METAPOST e Asymptote). Molto più semplicemente, li si può aggiungere al documento dopo averli preparati a parte con programmi specifici. I formati vettoriali più noti e diffusi sono il PDF, il PS e il suo parente stretto EPS e l'SVG, usato specialmente per le applicazioni Web.

Il paragrafo 6.4.4 nella pagina successiva raccoglie alcuni programmi per la grafica vettoriale

Immagini bitmap

Le immagini bitmap sono matrici di pixel colorati, di solito perdono in definizione se ingrandite o rimpicciolite e sono più adatte a fotografie e icone. I formati bitmap sono numerosissimi e vanno dal JPEG, molto diffuso in ambito fotografico e nella grafica con colori morbidi, al PNG, adatto per grafica con colori decisi, al GIF al TIFF. Alcuni di essi sono compressi per sfruttare al meglio la ridondanza d'informazione.

6.4.2 Convertire i formati

Prima ancora di includere le immagini nel documento bisogna produrle nel formato più adatto al proprio scopo. È inutile registrare una figura come JPEG per poi convertirla in PDF, perché la conversione include semplicemente il file bitmap in una "cornice" PDF senza migliorarne in alcun modo la qualità. È sbagliato anche fare l'opposto, perché così si perdono le informazioni sulla geometria della figura, abbassandone la qualità. Nonostante questo, si potrebbero avere a disposizione soltanto immagini in formati *non* adatti a LaTeX, e allora la conversione sarebbe davvero necessaria.

A questo proposito si ricordi bene che LaTeX accetta immagini PDF, JPEG, PNG e dalla versione 2011 di TeX Live anche EPS. Il pacchetto bmpsize, infine, permette di inserire anche immagini BMP, GIF e TIFF.

Il paragrafo 6.4.4 nella pagina seguente descrive alcuni programmi per convertire i diversi formati.

Tabella 40: Alcuni programmi utili per lavorare con LaTeX (le lettere G, C, R e V indicano rispettivamente le funzioni di grafica vettoriale, conversione dei formati, ritaglio immagini e visualizzazione; il simbolo • indica che la funzione è disponibile).

Programma	G	C	R	V
Inkscape	•	•		•
Xfig e WinFIG	•			
Gnuplot	•			
Mathematica	•			
OmniGraffle	•			
GIMP		•	•	•
Adobe Acrobat			•	•
Adobe Reader				•
Anteprima		•	•	•
ImageMagick		•		
`eps2pdf` e `epstopdf`		•		
Ghostview e GSview		•		•

6.4.3 Ritagliare le immagini

Uno dei parametri più importanti di una figura è l'informazione sulle dimensioni del rettangolo circoscritto a essa (*bounding box*). Questo contorno determina la grandezza effettiva dell'immagine e serve a LaTeX per calcolare lo spazio da riservarle sulla pagina. Idealmente, il contorno dovrebbe coincidere con il limite dell'immagine, ma talvolta le figure sono circondate da un invisibile bordo bianco più o meno spesso che genera non pochi problemi estetici: la figura appare sulla pagina troppo piccola o non centrata o circondata da eccessivi margini verticali, per esempio, anche se LaTeX la sta trattando nel modo corretto.

La primissima cosa da verificare, quindi, è che le dimensioni della *bounding box* siano corrette, aprendo la figura con un programma opportuno (come Adobe Reader o GIMP) e attivando la visualizzazione del contorno che, se scorretto, va ridimensionato. Se il problema riguarda poche figure si può risolvere a mano, ma se i file da ottimizzare sono molti, vanno corretti all'origine (magari configurando ad hoc il programma usato per produrli).

6.4.4 Alcuni programmi utili

La composizione asincrona di LaTeX presenta anche il non trascurabile vantaggio di assicurare sempre il prodotto migliore: per ciascuna operazione sul documento, infatti, l'autore può usare un programma specializzato. Questo non potrebbe accadere con un software "tuttofare" come un editor di testo tradizionale. La tabella 40 raccoglie, senza pretese di completezza, alcuni programmi utili per lavorare con LaTeX (ma preziosi anche in molte altre occasioni), specificandone le funzioni principali. Una veloce ricerca in Rete permette di recuperarli.

6.4.5 Includere le immagini nel documento

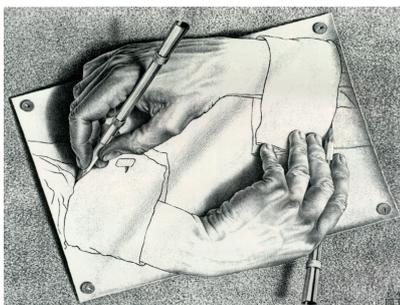
LaTeX gestisce le immagini con il pacchetto graphicx, che non richiede opzioni, da caricare nel modo consueto. Il comando `\includegraphics`, la cui sintassi completa è

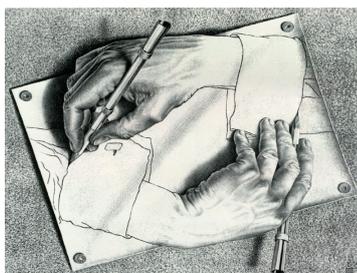
```
\includegraphics[⟨chiave⟩=⟨valore⟩,⟨...⟩]{⟨immagine⟩}
```

le include nel documento. Si osservi che:

- nell'argomento facoltativo ci vanno le opzioni che regolano l'aspetto della figura sulla pagina nella forma ⟨*chiave*⟩=⟨*valore*⟩ (si veda la tabella 41 nella pagina seguente);
- nell'argomento obbligatorio ci va il nome dell'immagine *senza* specificarne l'estensione.

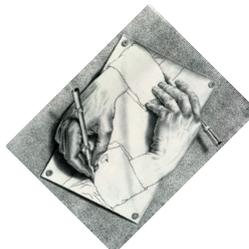
Gli esempi seguenti mostrano il pacchetto all'opera.
L'autore può assegnare all'immagine una larghezza (`width`)

```
\includegraphics[width=%
\textwidth]{Mani}
```

o un'altezza (`height`) a piacere:

```
\includegraphics[height=%
0.15\textheight]{Mani}
```

Ancora, la può ridimensionare nel suo complesso

```
\includegraphics[scale=0.10]%
{Mani}
```

o ruotare di un certo angolo in entrambi i sensi:

```
\includegraphics[width=0.5%
\textwidth,angle=45]{Mani}
```

Si noti che:

- le figure devono avere dimensioni *relative*: una frazione di `\textwidth` per la larghezza e di `\textheight` (cioè l'altezza della gabbia del testo) per l'altezza, per i motivi già spiegati nel paragrafo 6.1 a pagina 85;
- il valore di scalatura si esprime con un numero decimale;

Tabella 41: Principali chiavi di graphicx.

Chiave	Agisce su
`width`	Larghezza
`height`	Altezza
`angle`	Orientamento
`scale`	Larghezza e altezza

- l'angolo di rotazione si esprime con un numero (negativo, se la rotazione è oraria) nell'intervallo 1-360;
- se si intende assegnare all'immagine sia una larghezza sia un'altezza determinate, si ricordi di specificare anche la chiave `keepaspectratio` per evitare di distorcerla.

7 BIBLIOGRAFIA

Realizzare e gestire la bibliografia è da sempre uno degli aspetti più delicati nella stesura di un documento: LaTeX definisce tutti gli strumenti per svolgere questo compito con efficienza e flessibilità.

Si può creare la bibliografia in due modi, essenzialmente:

- a mano, con l'ambiente `thebibliography`;
- automaticamente, con il pacchetto biblatex.

7.1 BIBLIOGRAFIA MANUALE

L'ambiente `thebibliography` gestisce la bibliografia di un documento in modo molto facile, ma non altrettanto flessibile. La sintassi generale è:

```
\begin{thebibliography}{⟨etichetta più lunga⟩}
\bibitem[⟨etichetta personalizzata⟩]{⟨chiave di citazione⟩}
...
\end{thebibliography}
```

dove:

- ⟨*etichetta più lunga*⟩ può essere un numero (di solito 9 se la bibliografia comprende meno di dieci opere, 99 se almeno dieci ma meno di cento, eccetera) oppure del testo (nel caso di etichette personalizzate: si scrive allora l'etichetta più lunga);
- `\bibitem` va premesso a ogni voce bibliografica;
- ⟨*etichetta personalizzata*⟩ può contenere un'etichetta personalizzata che verrà stampata nella bibliografia e nelle citazioni al posto del numero predefinito;
- ⟨*chiave di citazione*⟩ è l'etichetta *univoca* per citare la fonte nel documento (si consiglia di usare la sintassi ⟨*autore*⟩:⟨*titolo*⟩ analoga a quella di `\label`).

Si vede l'ambiente all'opera nell'esempio seguente:

```
\begin{thebibliography}{9}
\bibitem{eco:tesi}
Eco, Umberto (1977),
\emph{Come si fa una tesi di
laurea}, Bompiani, Milano.

\bibitem{mori:tesi}
Mori, Lapo Filippo (2007),
“Scrivere la tesi di laurea
con \LaTeXe”, \Ars~(3).
\end{thebibliography}
```

BIBLIOGRAFIA

[1] Eco, Umberto (1977), *Come si fa una tesi di laurea*, Bompiani, Milano.

[2] Mori, Lapo Filippo (2007), "Scrivere la tesi di laurea con LaTeX 2_ε", ArsTeXnica (3).

Si noti che:

- di fatto, l'ambiente è identico a `itemize`;
- ciascuna voce bibliografica va scritta per intero, regolandone *a mano* tutti gli aspetti (corsivo, virgolette, eccetera), compresa la posizione nell'ordine alfabetico;
- nel documento finito le voci sono contrassegnate con un numero tra parentesi quadre sia nella bibliografia sia nelle citazioni;
- `thebibliography` produce la sezione contenente la bibliografia con la relativa testatina.

Per riferirsi a una fonte presente in bibliografia si usa il comando `\cite`:

```
\cite[⟨informazioni⟩]{⟨chiave di citazione⟩}
```

dove:

- ⟨*informazioni*⟩ sono le eventuali indicazioni di numero di pagina o di capitolo che si possono dare per completare la citazione;
- ⟨*chiave di citazione*⟩ si spiega da sé.

I prossimi esempi lo mostrano all'opera:

```
Si veda~\cite{eco:tesi} per
maggiori dettagli.
```

Si veda [1] per maggiori dettagli.

```
Si veda~\cite[p.~7]{eco:tesi}
per maggiori dettagli.
```

Si veda [1, p. 7] per maggiori dettagli.

Per far comparire nell'indice generale il titolo della sezione bibliografica del documento, *immediatamente prima* di aprire `thebibliography` vanno date sequenze di comandi diverse in base alla classe in uso:

```
\cleardoublepage
%\phantomsection
\addcontentsline{toc}{chapter}{\bibname}
```

se la classe è book o report, e

```
%\clearpage
%\phantomsection
\addcontentsline{toc}{section}{\refname}
```

se la classe è article, dove:

- `\cleardoublepage` fa cominciare la bibliografia in una pagina nuova dispari e assegna alla corrispondente voce nell'indice il numero di pagina corretto;
- `\clearpage` va dato per assicurare la corretta assegnazione del numero di pagina alla voce nell'indice *solo se* a fine composizione il corpo del documento terminasse esattamente a fine pagina e la bibliografia cominciasse perciò all'inizio di una pagina nuova (in tal caso si decommenti la riga corrispondente);
- `\phantomsection` va dato *solo se* è caricato anche hyperref (in tal caso si decommenti la riga corrispondente);
- `chapter` e `section` indicano il livello di sezionamento della sezione bibliografica (un capitolo e un paragrafo, rispettivamente);
- `\bibname` e `\refname` producono nell'indice generale del documento le voci *Bibliografia* e *Riferimenti bibliografici* rispettivamente.

7.2 BIBLIOGRAFIA AUTOMATICA

L'ambiente `thebibliography` produce risultati dignitosi, ma a prezzo di alcuni inconvenienti non da poco:

- l'ordinamento alfabetico delle fonti nella bibliografia è a carico dell'autore;
- bisogna scrivere una bibliografia per ogni documento, anche se c'è solo qualche fonte di differenza dall'uno all'altro;
- se si aggiorna il documento, bisogna modificare le bibliografie di *tutti* i documenti in cui esso è citato;
- per cambiare lo stile della bibliografia bisogna modificare a mano *tutte* le voci, una per una.

Come si noterà, anziché compilare bibliografie indipendenti è decisamente meglio generarle *automaticamente* creando *una volta per tutte* un solo database: sistemandolo in un'opportuna cartella dell'*albero personale* (per esempio, `$HOME/texmf/bibtex/bib/mieibib`) basterà aggiornarlo all'occorrenza.

In questo paragrafo vengono presentati gli elementi fondamentali del pacchetto biblatex, che offre all'autore una soluzione complessiva e automatica per gestire e personalizzare la bibliografia e gli stili bibliografici e di citazione in un documento composto con LaTeX. Infatti, oltre a includere in sé le funzioni di molti pacchetti, biblatex permette di effettuare modifiche anche profonde di un particolare stile bibliografico con una sola operazione.

Per il corretto funzionamento del pacchetto si raccomanda di caricare *anche* babel e csquotes (con l'opzione `babel`) rispettando la seguente sequenza:

```
\usepackage[⟨lingue⟩]{babel}
\usepackage[babel]{csquotes}
\usepackage[⟨opzioni⟩]{biblatex}
```

Si ricordi infine che biblatex *non* è compatibile con il pacchetto ucs.

7.2.1 Database bibliografici

Un *database bibliografico* è un file da registrare con estensione `.bib` (si scrive con l'editor in uso e *non* va composto) che contiene un certo numero di *record* come i seguenti:

```
@book{eco:tesi,
    author       = {Eco, Umberto},
    title        = {Come si fa una tesi di laurea},
    publisher    = {Bompiani},
    date         = {1977},
    location     = {Milano}
}

@article{mori:tesi,
     author       = {Mori, Lapo Filippo},
     title        = {Scrivere la tesi di laurea con \LaTeXe},
     journaltitle = {\Ars},
     number       = {3},
     date         = {2007}
}

@manual{mori:poesie,
     author       = {Mori, Lapo Filippo},
```

```
     title        = {Scrivere poesie con \LaTeX},
     date         = {2007},
     url          = {http://www.guitex.org/}
}

@online{wiki:latex,
     title        = {\LaTeX{} su Wikipedia},
     date         = {2011},
     url          = {http://it.wikipedia.org/wiki/LaTeX},
     sortkey      = {wiki},
     label        = {wiki}
}
```

Si osservi che:

- Ogni record corrisponde a una fonte, il cui tipo viene indicato come prima cosa (indifferentemente in maiuscolo o minuscolo) subito dopo il carattere `@`. Si tratta in questo caso di un libro (record di tipo `book`), di un articolo (`article`), di un manuale (`manual`) e di un sito Web (`online`).
- Si indica poi una *chiave*, di solito nella forma ⟨*cognome*⟩:⟨*parola chiave*⟩ (dove ⟨*parola chiave*⟩ è una parola caratterizzante la fonte), da mettere nell'argomento degli appositi comandi per citare la fonte nel documento.
- Infine, si riempiono i *campi* che definiscono l'opera (autore, titolo, eccetera). Ogni campo assume la forma

  ```
  ⟨nome del campo⟩ = {⟨contenuto del campo⟩}
  ```

  ed è separato dalla chiave e dagli altri campi con la virgola.

Per indicare a LaTeX quale database usare per comporre la bibliografia si dà nel preambolo il comando

```
\bibliography{⟨bibliografia1⟩,⟨bibliografia2⟩,⟨...⟩}
```

nel cui argomento si scrive il nome del file `.bib` *senza specificarne l'estensione*. Se le basi di dati da cui attingere sono più d'una, si scrivono separandole con la virgola *e senza spazi in mezzo*.

7.2.2 Record e campi

Ogni record contiene uno o più *campi*. Un campo può essere:

- *obbligatorio*, ovvero indispensabile a biblatex per generare la bibliografia;
- *opzionale*, cioè non indispensabile ma utile, se specificato, per aggiungere informazioni sulla fonte.

Record

Di seguito si elencano i principali *record* standard riconosciuti da biblatex, indicando per ognuno campi obbligatori e principali campi opzionali (il loro significato verrà spiegato nel paragrafo successivo). In particolari situazioni i campi in corsivo si possono sostituire con altri, come spiegato nel paragrafo 7.2.6 a pagina 120. I record e i campi disponibili con biblatex sono numerosissimi e rispondono alle esigenze più disparate; per il loro elenco completo si veda la documentazione del pacchetto.

article Articolo apparso in una rivista o in un giornale.
Campi obbligatori: author, title, journaltitle, date.
Campi opzionali: editor, series, volume, number, month, pages, note, url.

book Libro regolarmente pubblicato da una casa editrice.
Campi obbligatori: author, title, date.
Campi opzionali: editor, volume, series, note, publisher, location, number, url.

collection Raccolta di contributi indipendenti di autori diversi.
Campi obbligatori: editor, title, date.
Campi opzionali: volume, series, number, publisher, location, note, chapter, pages, url.

proceedings Lavori di una conferenza.
Campi obbligatori: *editor*, title, date.
Campi opzionali: volume, series, number, note, location, publisher, organization, chapter, pages, url.

inbook Parte di un libro con un titolo proprio.
Campi obbligatori: author, title, booktitle, date.
Campi opzionali: editor, volume, series, note, publisher, location, chapter, number, pages, url.

incollection Parte di una raccolta con un titolo proprio.
Campi obbligatori: author, title, booktitle, date.
Campi opzionali: volume, series, number, publisher, location, note, chapter, pages, url.

inproceedings Articolo nei lavori di una conferenza.
Campi obbligatori: author, editor, title, booktitle, date.
Campi opzionali: volume, series, number, note, location, publisher, organization, chapter, pages, url.

booklet Libro distribuito in proprio.
Campi obbligatori: author o editor, title, date.
Campi opzionali: howpublished, type, note, location, chapter, pages, url.

manual Documentazione tecnica.
Campi obligatori: *author* o *editor*, title, date.
Campi opzionali: type, version, series, number, organization, note, publisher, location, chapter, pages, url.

report Relazione pubblicata da università, scuola o altra istituzione.
Campi obbligatori: author, title, type, institution, date.
Campi opzionali: number, note, location, chapter, pages, url.

thesis Tesi di laurea o di dottorato. Il campo type specifica il tipo di tesi.
Campi obbligatori: author, title, type, institution, date.
Campi opzionali: note, location, chapter, pages, url.

online Risorsa disponibile su Internet.
Campi obbligatori: *author* o *editor*, title, date, url.
Campi opzionali: note, organization, date.

unpublished Documento con autore e titolo, ma non pubblicato.
Campi obbligatori: author, title, date.
Campi opzionali: howpublished, note, date, url.

misc Record da usare quando nessuno degli altri è appropriato.
Campi obbligatori: *author* o *editor*, title, date.
Campi opzionali: howpublished, type, organization, location, note, date, url.

Campi

Di seguito si riporta l'elenco dei principali *campi* riconosciuti da biblatex.

author Cognome e nome dell'autore (o degli autori, nel formato descritto più avanti) separati con la virgola.

booktitle Titolo dell'opera che contiene la fonte (se essa è solo una parte della pubblicazione).

chapter Numero del capitolo (o di una qualunque parte del documento).

editor Cognome e nome del curatore (o dei curatori) separati con la virgola.

howpublished Genere della pubblicazione.

institution Nome dell'università o dell'istituzione.

journaltitle Nome della rivista o del giornale.

label Etichetta per le citazioni, nel caso in cui manchino i dati necessari per formare l'etichetta "regolare" (si veda il paragrafo 7.2.6 a pagina 120).

location Indirizzo dell'editore (publisher) o dell'istituzione (institution).

month Mese di pubblicazione dell'opera. Dev'essere un numero intero; per esempio, non si scrive month={January}, ma month={1}.

note Informazioni supplementari che possono aiutare il lettore a identificare meglio l'opera.

number Numero della rivista, del giornale o della raccolta.

organization Organizzazione che pubblica il documento o patrocina la conferenza.

pages Uno o più numeri di pagina.

publisher Nome dell'editore.

series Nome della collezione di cui la fonte fa parte.

sortkey Ordina alfabeticamente le voci bibliografiche prive dell'indicazione di autore o curatore (si veda il paragrafo 7.2.6 a pagina 120).

title Titolo della fonte.

type Tipo di manuale, relazione o tesi.

url Indirizzo Internet di riferimento per la fonte.

volume Numero di volume della fonte.

date Anno di pubblicazione della fonte.

Alcune precisazioni

Gli stili *predefiniti* di biblatex stampano il titolo dell'opera rispettando le maiuscole e le minuscole. Un titolo come

```
title = {TCP-IP e lo Zen di Confucio}
```

verrà stampato esattamente com'è scritto. Alcuni suoi stili *personalizzati*, invece, lo stampano automaticamente tutto in minuscolo, con risultati a volte indesiderati. Si può risolvere il problema racchiudendo in ulteriori gruppi di graffe solo le parole problematiche, così:

```
title = {{TCP-IP} e lo {Zen} di {Confucio}}
```

oppure l'intero titolo, così:

```
title = {{TCP-IP e lo Zen di Confucio}}
```

Un comando che produce testo presente in un campo non va racchiuso tra graffe se si lavora con uno stile predefinito:

```
title = {L'arte di scrivere con \LaTeX}
```

ma potrebbe essere necessario farlo se lo stile è personalizzato:

```
title = {L'arte di scrivere con {\LaTeX}}
```

Più nomi in un campo `author` o `editor` vanno separati con `and`:

```
author = {Mori, Lapo Filippo and Himmelmann, Maurizio}
```

Se l'elenco dei nomi degli autori o dei curatori è troppo lungo, può essere concluso da `and others`, che di regola viene reso da biblatex con *et al.*:

```
author = {Gregorio, Enrico and Mori, Lapo Filippo and
          Pantieri, Lorenzo and others}
```

I cognomi multipli si scrivono separandoli dal nome in questo modo:

```
author = {Levi Montalcini, Rita}
```

Con i cognomi preceduti da particelle con iniziale *minuscola* (*von* o *van*, per esempio) biblatex si comporta in modo particolare. Si immagini che l'autore sia *Ludwig van Beethoven*: gli stili standard considerano il cognome come se non avesse la particella e lo mettono in ordine alfabetico di conseguenza. Per evitarlo, seguendo la tradizione italiana, basta dare a biblatex l'opzione `useprefix`:

```
\usepackage[⟨...⟩,useprefix]{biblatex}
```

e il cognome viene considerato per intero. (Con qualunque stile, l'opzione `useprefix=false`, predefinita negli stili standard, tratta il cognome come se non avesse la particella davanti.)

Se il cognome è invece preceduto da una particella con iniziale maiuscola, viene sempre considerato per intero e messo in ordine alfabetico di conseguenza. Per esempio, *De Gasperi* cade sotto la lettera *d*.

7.2.3 Stili bibliografici e schemi di citazione

L'aspetto delle voci nella bibliografia finale e delle citazioni nel corpo del documento, che biblatex adatta automaticamente alla lingua principale in uso, si specificano in modi diversi. Il codice

```
\usepackage[bibstyle=authortitle,%
            citestyle=verbose-trad1]{biblatex}
```

imposta lo *stile bibliografico* (`bibstyle`) autore-titolo (`authortitle`) per la bibliografia finale e uno stile di citazione (`citestyle`) che nella prima citazione del documento riporta il riferimento bibliografico per intero (`verbose-trad1`) per le citazioni nel documento, mentre il codice

```
\usepackage[style=alphabetic]{biblatex}
```

imposta con lo stile alfabetico (`alphabetic`) entrambi i parametri. È il tipo di lavoro che si sta scrivendo a determinare lo schema (e quindi lo stile) più adatto da usare.

Stili bibliografici

Il pacchetto biblatex fornisce quattro *stili bibliografici* predefiniti, diversi nel formato complessivo dell'elenco dei riferimenti bibliografici e nel tipo di etichetta che ci viene stampato. Tutti e quattro elencano le fonti nella bibliografia secondo l'ordine alfabetico del cognome del loro autore. Di seguito li si descrivono brevemente.

`numeric` Assegna a ciascuna fonte un numero progressivo ([1], [2], eccetera). Questo stile è diffusamente usato nei documenti scientifici, dove spesso il risultato cui ci si riferisce è più importante di chi ha scritto la fonte che lo contiene e quando.

`authoryear` Non prevede alcuna etichetta e mette l'anno di pubblicazione della fonte subito dopo il nome dell'autore (o del curatore). Lo stile è diffuso nei documenti umanistici (ma solo se la bibliografia è *moderna* [Eco, 1977]) dove è importante indicare autore e anno di pubblicazione della fonte direttamente nel testo.

`alphabetic` L'etichetta è formata dalla prima parte del cognome dell'autore e dalle ultime due cifre dell'anno di pubblicazione ([Mor07]). Sta a metà strada tra i due stili appena descritti, ed è utile se l'autore deve sottostare a precisi standard editoriali oppure preferisce una via di mezzo.

`authortitle` Non prevede alcuna etichetta. Questo schema (per cui un'opera citata nel testo ha il proprio riferimento in una nota al piede) si usa esclusivamente nei documenti umanistici: il lettore conosce subito l'opera cui lo si rimanda, ma la pagina potrebbe diventare troppo piena di note e perciò faticosa da leggere.

Schemi di citazione

A ciascuno stile si possono associare (non contemporaneamente) uno o più *schemi di citazione* appropriati, che ne riproducono l'effetto nel corpo del documento. Ogni schema, però, può essere associato a *un solo* stile bibliografico. Di seguito si descrivono quattro degli schemi predefiniti da biblatex e alcune loro varianti, utili per ottenere riferimenti in altri formati.

`numeric`, `numeric-comp` Riferimento: numerico ([1], [2], eccetera).
Da associare a: stile `numeric`.
Varianti: `numeric-comp`, che ordina e comprime automaticamente le citazioni multiple. Per esempio, si ottiene [2–4, 8] al posto di [4, 2, 8, 3].

Bibliografia

[1] Umberto Eco. *Come si fa una tesi di laurea.* Milano: Bompiani, 1977.
[2] Lapo Filippo Mori. "Scrivere la tesi di laurea con LATEX 2ε". In: *ArsTEXnica* 3 (2007).
[3] Lapo Filippo Mori. *Scrivere poesie con LATEX.* 2007. URL: `http://www.guit.sssup.it/`.
[4] *LATEX su Wikipedia.* 2010. URL: `http://it.wikipedia.org/wiki/LaTeX`.

(a) Stile `numeric`.

Bibliografia

[Eco77] Umberto Eco. *Come si fa una tesi di laurea.* Milano: Bompiani, 1977.
[Mor07a] Lapo Filippo Mori. "Scrivere la tesi di laurea con LATEX 2ε". In: *ArsTEXnica* 3 (2007).
[Mor07b] Lapo Filippo Mori. *Scrivere poesie con LATEX.* 2007. URL: `http://www.guit.sssup.it/`.
[Wiki10] *LATEX su Wikipedia.* 2010. URL: `http://it.wikipedia.org/wiki/LaTeX`.

(b) Stile `alphabetic`.

Bibliografia

Eco, Umberto (1977). *Come si fa una tesi di laurea.* Milano: Bompiani.
Mori, Lapo Filippo (2007a). "Scrivere la tesi di laurea con LATEX 2ε". In: *ArsTEXnica* 3.
– (2007b). *Scrivere poesie con LATEX.* URL: `http://www.guit.sssup.it/`.
Wiki (2010). *LATEX su Wikipedia.* URL: `http://it.wikipedia.org/wiki/LaTeX`.

(c) Stile `authoryear`.

Bibliografia

Eco, Umberto. *Come si fa una tesi di laurea.* Milano: Bompiani, 1977.
Mori, Lapo Filippo. "Scrivere la tesi di laurea con LATEX 2ε". In: *ArsTEXnica* 3 (2007).
– *Scrivere poesie con LATEX.* 2007. URL: `http://www.guit.sssup.it/`.
LATEX su Wikipedia. 2010. URL: `http://it.wikipedia.org/wiki/LaTeX`.

(d) Stile `authortitle`.

Figura 7: Esempi di stili bibliografici.

alphabetic Riferimento: misto ([Mor07]).
Da associare a: stile alphabetic.

authoryear, **authoryear-comp** Riferimento: autore, anno ([Mori, 2007]).
Da associare a: stile authoryear.
Varianti: authoryear-comp, che ordina e comprime automaticamente le citazioni multiple con lo stesso autore (ed eventualmente lo stesso anno di pubblicazione). Per esempio, si ottiene [Eco 1977; Mori 2007a,b] anziché [Mori 2007b; Eco 1977; Mori 2007a].

authortitle, **authortitle-comp**, **verbose**, **verbose-trad1** Riferimento: autore, titolo (Mori, *Titolo*).
Da associare a: stile authortitle.
Varianti:

- authortitle-comp, che ordina e comprime automaticamente le citazioni multiple di opere con lo stesso autore. Per esempio, si ottiene [Mori, *Primo titolo*, *Secondo titolo*] anziché [Mori, *Primo titolo*; Mori, *Secondo titolo*].
- verbose, che usa la citazione completa la prima volta e una sua forma abbreviata le volte successive; prevede a sua volta alcune varianti (tra cui verbose-trad1) in cui vengono usate, a seconda del contesto, le formule convenzionali *idem*, *ibidem*, *op. cit.* e *loc. cit.*

Per l'elenco completo degli schemi di citazione di biblatex si rimanda alla documentazione del pacchetto.

Per ogni schema di citazione c'è uno stile bibliografico adatto, che biblatex imposta automaticamente in base allo schema scelto. Il codice

```
\usepackage[style=authoryear-comp]{biblatex}
```

infatti, equivale a

```
\usepackage[bibstyle=authoryear,%
            citestyle=authoryear-comp]{biblatex}
```

Oltre a quelli predefiniti si possono usare molti altri stili. La bibliografia di questa guida, per esempio, è composta con lo stile philosophy-modern, presente in TEX Live. In Rete, infine, se ne possono trovare numerosi altri per comporre bibliografie conformi a precisi standard: su ☞ CTAN ce ne sono alcuni.

7.2.4 Comandi per le citazioni

Una volta creati i database, il più è fatto. Ora ci si può riferire a una fonte presente in bibliografia con gli appositi comandi. Quello più usato è \cite, che si usa come spiegato nel paragrafo 7.1 a pagina 109.

Il comando \nocite, invece, permette di far comparire nella bibliografia finale anche fonti non esplicitamente citate nel documento:

- solo alcune, se dato nel corpo del documento al posto di \cite (come mostra il codice che produce la figura 8a a pagina 121);
- tutte, se dato prima di \printbibliography (si veda più oltre) nella forma \nocite{*}.

Oltre ai due comandi appena visti, biblatex ne definisce altri per citazioni ancora diverse:

- \textcite quando la citazione è parte integrante del discorso;

- `\parencite` racchiude la citazione fra parentesi;
- `\footcite` mette la citazione in una nota;
- `\supercite` (solo per schemi numerici) mette la citazione in apice;
- `\fullcite` mette come citazione l'intera voce bibliografica.

Si noti che l'opzione `natbib` di biblatex permette di mantenere una compatibilità quasi totale con i documenti che caricano natbib: la maggior parte dei nomi dei comandi per le citazioni di natbib, come `\citep` e `\citet`, sono conservati come alias.

Di seguito si mostrano gli stessi esempi d'uso di `\cite` con lo schema autore-anno compatto:

```
\cite{eco:tesi} \\
\cite[5]{eco:tesi} \\
\cite[5-9]{eco:tesi} \\
\cite[vedi][]{eco:tesi} \\
\cite[vedi][5]{eco:tesi}
```

Eco, 1977
Eco, 1977, p. 5
Eco, 1977, pp. 5-9
vedi Eco, 1977
vedi Eco, 1977, p. 5

e con lo schema numerico compatto:

```
\cite{eco:tesi} \\
\cite[5]{eco:tesi} \\
\cite[5-9]{eco:tesi} \\
\cite[vedi][]{eco:tesi} \\
\cite[vedi][5]{eco:tesi}
```

[1]
[1, p. 5]
[1, pp. 5–9]
[vedi 1]
[vedi 1, p. 5]

(in modo del tutto analogo funzionano gli altri comandi, si facciano delle prove). Come si può osservare, biblatex produce automaticamente le abbreviazioni *p.* e *pp.* a seconda delle circostanze. (Si può ottenere anche la scrittura estesa *pagina* o *pagine*: si veda la documentazione del pacchetto.)

Per citare singole parti di una voce bibliografica ci sono `\citeauthor` e `\citeyear`:

```
\citeauthor{eco:tesi} \\
\citeyear{eco:tesi}
```

Eco
1977

Lo schema `authoryear-comp` mette le citazioni multiple in ordine alfabetico:

```
\parencite{mori:poesie,%
   eco:tesi,mori:tesi}
```

(Eco, 1977; Mori 2007a,b)

Lo schema `numeric-comp` ordina e comprime le citazioni multiple:

```
\parencite{mori:poesie,%
   eco:tesi,mori:tesi}
```

[1–3]

7.2.5 Generare e collocare la bibliografia nel documento

Il comando `\printbibliography` produce la sezione contenente la bibliografia con la relativa testatina. Va dato *immediatamente dopo* le stesse sequenze di comandi già viste nel paragrafo 7.1 a pagina 109 per mandarne il relativo titolo nell'indice generale:

```
\cleardoublepage
%\phantomsection
\addcontentsline{toc}{chapter}{\bibname}
\printbibliography
```

se la classe è book o report, e

```
%\clearpage
%\phantomsection
\addcontentsline{toc}{section}{\refname}
\printbibliography
```

se la classe è article. Anche per biblatex valgono le stesse osservazioni contenute in quel paragrafo, cui si rimanda.

Per generare effettivamente la bibliografia nel documento, infine, questa è la sequenza di composizione:

1. si compone il documento con LaTeX una prima volta;
2. si lancia il programma BibTeX premendo l'apposito pulsante dell'editor;
3. si compone il documento altre *due* volte con LaTeX.

7.2.6 Specialità

In questo paragrafo vengono presentati alcuni argomenti avanzati relativi a biblatex.

Campi speciali

I campi elencati in questo paragrafo non contengono dati stampabili, ma servono per scopi diversi e si possono applicare a *tutti* i record bibliografici.

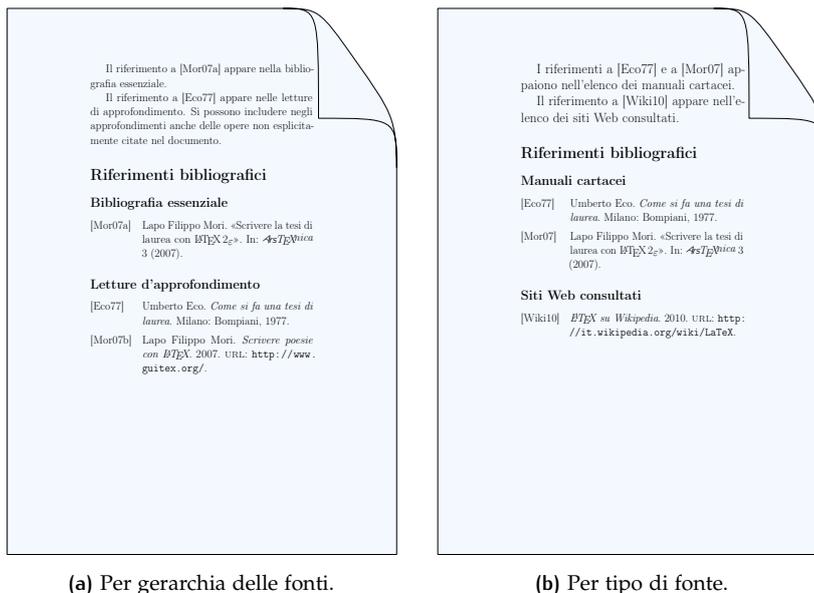
hyphenation Imposta la lingua della voce bibliografica. Il valore deve essere il nome di una lingua nota a babel.

sortkey Imposta l'ordine alfabetico degli elementi della bibliografia: serve come chiave di ordinamento nei record privi dell'indicazione di autore o curatore.

keywords Un elenco di parole chiave separate da virgole, che fanno da *filtro* sulle voci bibliografiche da stampare (per un esempio, si veda il paragrafo 7.2.6 a fronte).

Campi omissibili

Alcuni dei campi indicati come obbligatori nel paragrafo 7.2.2 a pagina 112 in realtà non lo sono sempre. Nella bibliografia del proprio lavoro ci possono essere un libro anonimo oppure la raccolta dei lavori di una conferenza senza un curatore. Di regola questo non è un problema nella composizione della bibliografia, ma può esserlo nelle citazioni: uno schema di citazione autore-anno richiede *sempre* un campo author/editor e un campo date.

Se mancano i dati necessari per formare l'etichetta "regolare", si può sostituire ogni dato mancante con il campo label, da usare nei modi propri di ogni schema di citazione. Negli schemi autore-anno sostituisce il campo author/editor oppure il campo date nelle opere che ne sono prive. Negli schemi numerici, invece, non viene usato, perché in questo caso il formato delle citazioni è indipendente dai dati della bibliografia. Lo stesso accade negli schemi autore-titolo, perché il solo titolo basta di solito per formare una citazione univoca (di regola, ogni fonte ha un titolo).

(a) Per gerarchia delle fonti.

(b) Per tipo di fonte.

Figura 8: Bibliografie suddivise.

Riferimenti finali, cliccabili e bibliografia multilingue

Le seguenti opzioni permettono di personalizzare la bibliografia del proprio documento:

`backref` indica accanto a ciascuna voce bibliografica le pagine del documento in cui la fonte è citata;

`hyperref` (che richiede il pacchetto omonimo) rende cliccabili le citazioni e i riferimenti finali;

`babel=hyphen` ordina a LaTeX di verificare per quale o per quali voci della bibliografia si è specificata una lingua diversa da quella corrente e di sillabarla con le regole della nuova lingua;

`babel=other` dice a LaTeX di mettere la voce in un ambiente `otherlanguage` e di tradurre nella nuova lingua anche tutti i termini associati all'opera, come *curatore*, *volume*, eccetera.

Di seguito si mostra un record in cui si è specificata la lingua:

```
@book{lamport:latex,
    author      = {Lamport, Leslie},
    title       = {\LaTeX. A Document Preparation System},
    publisher   = {Addison-Wesley},
    year        = {1994},
    location    = {Reading (Massachusetts)},
    hyphenation = {english}
}
```

Suddividere la bibliografia

Talvolta è necessario suddividere la bibliografia in base a certi criteri. Per esempio, la si può ripartire in letteratura di base e di approfondimento, oppure elencare separatamente le risorse cartacee e quelle online, oppure ancora separare le opere citate esplicitamente nel documento da quelle che

s'intendono semplicemente suggerire. Con biblatex tutto questo si può fare facilmente.

Si supponga di aver compilato un database bibliografico `Bibliografia.bib` secondo le istruzioni contenute nel paragrafo 7.2.1 a pagina 111. Si consideri il seguente codice:

```
\documentclass{article}
\usepackage[italian]{babel}
\usepackage[babel]{csquotes}
\usepackage[style=alphabetic]{biblatex}
\usepackage{guit}
\bibliography{Bibliografia}

\DeclareBibliographyCategory{basi}
\DeclareBibliographyCategory{approf}

\addtocategory{basi}{mori:tesi}
\addtocategory{approf}{eco:tesi,mori:poesie}

\defbibheading{basi}{\subsection*{Bibliografia essenziale}}
\defbibheading{approf}{\subsection*{Letture d'approfondimento}}

\begin{document}

Il riferimento a~\cite{mori:tesi} appare nella bibliografia
essenziale.

Il riferimento a~\cite{eco:tesi} appare nelle letture
di approfondimento. Si possono includere negli approfondimenti
anche delle opere non esplicitamente citate nel documento.
\nocite{mori:poesie}

\section*{\refname}
\printbibliography[heading=basi,category=basi]
\printbibliography[heading=approf,category=approf]

\end{document}
```

Si tratta di un articolo con due bibliografie separate. Si osservi ora quanto segue.

- Il comando `\DeclareBibliographyCategory` nel preambolo definisce due *categorie bibliografiche*: una per la letteratura di base, l'altra per gli approfondimenti. Le fonti vengono assegnate a ciascuna categoria con `\addtocategory`.
- Il comando `\defbibheading` definisce il titolo delle due bibliografie (*Bibliografia essenziale* e *Letture d'approfondimento*), che vengono composte esattamente come due sottoparagrafi (è la prassi, in un articolo). (Scrivendo un book o un report con bibliografia suddivisa, nel codice precedente `\subsection*` e `\section*{\refname}` vanno sostituiti con `\section*` e `\chapter*{\bibname}` rispettivamente.)
- L'argomento facoltativo di `\printbibliography` specifica le istruzioni di controllo: `heading=`⟨*nome*⟩ imposta il titolo della bibliografia come specificato con `\defbibheading`, `category=`⟨*categoria*⟩ *filtra* le voci da stampare, selezionando soltanto quelle che appartengono alla ⟨*categoria*⟩ specificata.

Al posto dei due `\printbibliography` con i relativi argomenti facoltativi si può scrivere un semplice

```
\bibbycategory
```

che equivale a dare un `\printbibliography` per ogni categoria nell'ordine in cui le si è dichiarate. Componendo il codice precedente si ottiene il risultato della figura 8a a pagina 121.

Un altro esempio. Si supponga di partire ancora dallo stesso database bibliografico appena considerato e si osservi il codice seguente:

```
\documentclass{article}
\usepackage[italian]{babel}
\usepackage[babel]{csquotes}
\usepackage[style=alphabetic]{biblatex}
\usepackage{guit}
\bibliography{Bibliografia}

\defbibheading{cartaceo}{\subsection*{Manuali cartacei}}
\defbibheading{web}{\subsection*{Siti Web consultati}}

\begin{document}

I riferimenti a~\cite{eco:tesi} e a~\cite{mori:tesi}
appaiono nell'elenco dei manuali cartacei.

Il riferimento a~\cite{wiki:latex} appare nell'elenco dei siti Web
consultati.

\section*{\refname}
\printbibliography[heading=cartaceo,nottype=online]
\printbibliography[heading=web,type=online]

\end{document}
```

Si tratta di un articolo con due bibliografie separate, una per i manuali cartacei e l'altra per i siti Web consultati (una "sitografia"). Si osservi che:

- il comando `\defbibheading` definisce il titolo delle due bibliografie (*Manuali cartacei* e *Siti Web consultati*);
- l'argomento facoltativo di `\printbibliography` filtra le voci come prima, per cui `type=`⟨*tipo*⟩ stampa soltanto quelle il cui campo `type` è ⟨*tipo*⟩, mentre `nottype=`⟨*tipo*⟩ stampa soltanto i record il cui campo `type` *non* è ⟨*tipo*⟩.

Componendo il codice precedente si ottiene il risultato della figura 8b a pagina 121.

8 INDICE ANALITICO

L'indice analitico è un elenco alfabetico di parole o espressioni (anche con alcuni livelli di subordinazione) dette *voci*, posto di regola alla fine di un documento; vicino a ogni voce compare l'elenco dei numeri delle pagine in cui la voce in questione è menzionata. In molti lavori un indice di questo tipo costituisce una sezione di enorme utilità, e LaTeX è capace di gestirlo automaticamente e con grande efficienza, come si mostra in questo capitolo.

Per creare l'indice analitico con LaTeX bisogna innanzitutto eseguire due operazioni preliminari:

1. caricare il pacchetto `makeidx` come di consueto, per abilitare il programma alla composizione dell'indice;
2. dare il comando `\makeindex` *nel preambolo*, per attivare i comandi dedicati che verranno inseriti nel corpo del testo.

A questo punto, *immediatamente dopo* ogni parola o espressione che si desidera indicizzare basta dare il comando

```
\index{⟨voce⟩}
```

Per produrre la sezione dell'indice analitico nel documento e mandarne il relativo titolo nell'indice generale, *immediatamente prima* di `\end{document}` vanno date le stesse sequenze di comandi viste per la bibliografia nel paragrafo 7.1 a pagina 109:

```
\cleardoublepage
%\phantomsection
\addcontentsline{toc}{chapter}{\indexname}
\printindex
```

se la classe in uso è book o report, e

```
%\clearpage
%\phantomsection
\addcontentsline{toc}{section}{\indexname}
\printindex
```

se la classe è article. Il codice

```
\documentclass{article}
...
\usepackage{makeidx}
\makeindex
...
\begin{document}

Si possono inserire nell'indice analitico singole parole
come \emph{arte}\index{arte}, oppure intere espressioni
come questa\index{intere espressioni come questa}.

...

\addcontentsline{toc}{section}{\indexname}
\printindex

\end{document}
```

Tabella 42: Sintassi del comando `\index`.

Tipo di voce e codice	Risultato
Primaria `\index{Artisti}`	Artisti, 2
Sottolemma `\index{Artisti!Escher}`	Artisti, 2 Escher, 3
Voce in altro stile `\index{Gaudì@\textit{Gaudì}}`	*Gaudì*, 5
Pagina in altro stile `\index{Klimt|textbf}`	Klimt, **7**
Rimando `\index{Picasso|see{Cubismo}}`	Picasso, *si veda* Cubismo
Intervallo di pagine `\index{Mirò|(}` ...`\index{Mirò|)}`	Mirò, 14–18

crea un indice analitico con le voci *arte* e *intere espressioni come questa*.

Per generare effettivamente l'indice analitico, questa è la sequenza di composizione da seguire:

1. si compone il documento con LaTeX una prima volta;
2. si lancia il programma `MakeIndex` premendo l'apposito pulsante dell'editor;
3. si compone il documento altre *due* volte con LaTeX.

Si ricorda infine il pacchetto imakeidx, con cui si possono ottenere automaticamente indici multipli inseriti nel documento *già alla prima composizione*.

9 PERSONALIZZAZIONI

Chiunque scriva con LaTeX ha bisogno, prima o poi, di "cose" non contemplate da LaTeX standard, e perciò si crea comandi e ambienti ad hoc o modifica quelli che già ci sono. In questo capitolo si impara come farlo.

9.1 COMANDI E AMBIENTI PERSONALI

9.1.1 Nuovi comandi

S'immagini di scrivere un libro di botanica e di volere tutti i nomi scientifici di pianta in corsivo, come si fa di solito. Il modo più immediato per farlo è scrivere ogni nome nell'argomento di `\textit`. Successivamente, per scelte tipografiche imperscrutabili, l'editore vi chiede i nomi di pianta in nero. Che fare? Si potrebbero sostituire automaticamente tutti i `\textit` con altrettanti `\textbf`, ma ci sarebbe qualche problema, perché potreste aver usato il corsivo anche per altre parti del libro, che debbono invece rimanere tali. Non rimane che sostituire un comando dopo l'altro e perdere un sacco di tempo.

LaTeX risolve questi problemi molto più "filosoficamente" di quanto non facciano altri programmi. Anziché agire a mano, si può definire *una volta per tutte* un nuovo comando, che si chiamerà `\pianta`, che stampa il proprio argomento (un nome di pianta, in questo caso) nello stile che l'autore decide (in corsivo, in questo caso). Se l'imposizione del nero avviene a documento completato, allora si modifica *una volta per tutte* la definizione del comando e non più tutti i `\pianta` nel documento.

Il nuovo comando, dunque, non descrive come l'argomento debba essere *stampato*, ma come lo si è *pensato*. Perciò in un documento scritto con LaTeX i comandi per cambiare lo stile vanno usati molto di rado, in favore di comandi che rispecchino la logica di ciò che si sta scrivendo.

I comandi personalizzati si definiscono *nel preambolo* con `\newcommand`:

```
\newcommand{⟨nome⟩}[⟨numero di argomenti⟩]{⟨definizione⟩}
```

Dove:

- ⟨*nome*⟩ è il nome che si dà al nuovo comando.

- ⟨*numero di argomenti*⟩ è il numero di argomenti obbligatori che si assegna al nuovo comando, fino a un massimo di nove. Se non si specifica questo valore, allora il nuovo comando non ha alcun argomento.

- ⟨*definizione*⟩ contiene le istruzioni che specificano ciò che si vuole che il nuovo comando "faccia". Se gli si sono assegnati degli argomenti, nella ⟨*definizione*⟩ questi vengono indicati con # seguiti dal loro numero progressivo: `#1`, `#2`, eccetera.

Il comando appena esaminato serve a definire nuovi comandi senza e con argomenti. L'esempio seguente mostra la sintassi di un comando personalizzato *senza argomenti*:

```
\newcommand{\arte}{\emph{L'arte di scrivere con \LaTeX}}
```

utile, per esempio, se in un documento si dovesse scrivere ripetutamente il titolo di questa guida (si noti che un comando di questo tipo "produce testo", e va perciò spaziato da ciò che lo segue nei modi già visti nel paragrafo 3.4.4 a pagina 24):

```
Questo lavoro s'intitola \arte.
```

Questo lavoro s'intitola *L'arte di scrivere con LaTeX*.

Si faccia attenzione, però, a non esagerare con i comandi personali semplicemente per abbreviare il testo, come `\gb` per *Gran Bretagna*, per esempio: alla lunga rendono il codice illeggibile.

Il prossimo codice, invece, mostra la sintassi di un comando personalizzato *con argomenti*.

```
\newcommand{\pianta}[1]{\textit{#1}}
```

Si è assegnato al nuovo comando un solo argomento obbligatorio (`[1]`). Quando si dà il comando, accade questo: il testo nell'argomento di `\pianta` viene "passato" a `#1` e trattato secondo la ⟨*definizione*⟩. In questo caso verrà stampato in corsivo, come si vede nell'esempio seguente:

```
\pianta{Rosa canina}
```

Rosa canina

In modo analogo, anche se accade davvero di rado, si possono definire nuovi comandi con più argomenti (specificando `[2]`, `[3]`, eccetera).

LaTeX impedisce di definire un comando già esistente, ma non di ridefinirlo: per farlo, si usa `\renewcommand`, che ha la stessa sintassi di `\newcommand`.

A chi volesse approfondire gli aspetti della programmazione avanzata di LaTeX si consiglia la lettura di [Gregorio, 2009].

9.1.2 Nuovi ambienti

Gli ambienti personalizzati si definiscono *nel preambolo* con il comando `\newenvironment`:

```
\newenvironment{⟨nome⟩}[⟨numero di argomenti⟩]%
   {⟨comandi di apertura⟩}{⟨comandi di chiusura⟩}
```

Dove:

- ⟨*nome*⟩ è il nome che si dà al nuovo ambiente.
- ⟨*numero di argomenti*⟩ è il numero di argomenti che gli si assegna. Se non si specifica questo valore, allora il nuovo ambiente non ha alcun argomento.
- ⟨*comandi di apertura*⟩ contiene le istruzioni da dare all'inizio dell'ambiente, che fanno uso degli argomenti eventualmente presenti.
- ⟨*comandi di chiusura*⟩ contiene le istruzioni da dare alla chiusura dell'ambiente.

In pratica, definire un nuovo ambiente equivale a definire due nuovi comandi, uno per la sua apertura e uno per la sua chiusura, da usare nel modo consueto.

L'esempio seguente mostra la sintassi di un ambiente personalizzato *senza argomenti*. Scrivendo nel preambolo

```
\newenvironment{itaitemize}{\begin{itemize}\itshape}{\end{itemize}}
```

si potrà usare il nuovo ambiente `itaitemize` così:

```
\begin{itaitemize}
\item Un elenco con voci\dots
\item \dots automaticamente
      in corsivo.
\end{itaitemize}
```

- *Un elenco con voci...*
- *...automaticamente in corsivo.*

Si noti che gli eventuali argomenti assegnati al nuovo ambiente possono essere assegnati soltanto ai ⟨*comandi di apertura*⟩. Per ridefinire ambienti già esistenti, si usa `\renewenvironment` (analogo a `\renewcommand`), che ha la stessa sintassi di `\newenvironment`.

Chi volesse cimentarsi nella costruzione di interi pacchetti o classi personalizzati, trova le istruzioni nel file `clsguide.pdf`, recuperabile come di consueto.

9.2 PERSONALIZZARE IL TESTO

9.2.1 Font

In tipografia la parola *font* indica un insieme di caratteri dei più svariati tipi (detti anche *glifi*) accomunati da un certo stile grafico. Più font con caratteristiche comuni costituiscono una *famiglia di font*, che comprende senz'altro la variante regolare (detta anche *tondo*) e almeno il *corsivo*, il MAIUSCOLETTO e il **nero**. Tra le caratteristiche più importanti di un font ci sono la *larghezza* e la presenza o meno delle *grazie*.

I font `a larghezza fissa` (o *monospaced* o *typewriter*, usati nei codici d'esempio di questa guida) presentano caratteri la cui larghezza è uguale per tutti e sono adatti per scrivere in un editor di testi o per incolonnare dati numerici, caratteristica che non hanno i font a larghezza *variabile*, più versatili e indicati per riempire la riga in modo ottimale.

I caratteri di un font *con grazie* (o *serif*, usati nel testo principale) presentano piccole rifiniture alle estremità che alla lunga li rendono più leggibili su carta, al contrario di quelli di un font senza grazie (o *sans serif*, usati per i nomi dei pacchetti), più indicati per la lettura a schermo o per corpi molto piccoli. Le figure 9 e 10 nella pagina successiva mostrano le quattro categorie di font appena esaminate.

Di regola, LaTeX sceglie la variante appropriata in base alla struttura logica del documento (capitoli, paragrafi, testatine, eccetera). Talvolta, però, si ha la necessità di modificarne a mano stile e corpo.

9.2.2 Modificare stile e corpo del font in uso

Modificare lo stile

I comandi elencati nella tabella 43 a pagina 131 modificano lo *stile* del loro argomento (e solo di quello), lasciando invariato il testo seguente.

```
La parola che segue è in
\textit{corsivo}.
Il resto del testo è normale.
```

La parola che segue è in *corsivo*. Il resto del testo è normale.

I comandi si possono combinare, ma il risultato dipende dalla presenza o meno nella distribuzione del font disegnato nello stile richiesto, come mostrano gli esempi seguenti:

(a) Con grazie: Times.

(b) Senza grazie: Helvetica.

Figura 9: Famiglie di font a larghezza variabile.

(a) Con grazie: Courier.

(b) Senza grazie: Monaco.

Figura 10: Famiglie di font a larghezza fissa.

Tabella 43: Comandi e dichiarazioni per modificare lo stile del font. Per evidenziare la differenza dal corsivo, lo stile inclinato è composto con il font Computer Modern.

Comando	Dichiarazione	Stile
`\emph`	`\em`	*Evidenziato*
`\textit`	`\itshape`	*Corsivo*
`\textsc`	`\scshape`	MAIUSCOLETTO
`\textbf`	`\bfseries`	**Nero**
`\textsl`	`\slshape`	*Inclinato*
`\textrm`	`\rmfamily`	Tondo
`\textsf`	`\sffamily`	Senza grazie
`\texttt`	`\ttfamily`	`Macchina per scrivere`

```
L'espressione che segue
\textit{\textbf{è in nero
corsivo}}, ma questa
\textsc{\textit{non è in
maiuscoletto corsivo}}.
```

L'espressione che segue ***è in nero corsivo***, ma questa *non è in maiuscoletto corsivo*.

A ciascuno di questi comandi corrisponde una dichiarazione che si comporta come spiegato nel paragrafo 3.4.4 a pagina 24. Come i comandi, si possono combinare anche le dichiarazioni:

```
L'espressione che segue
{\itshape è in {\bfseries
nero corsivo}}.
```

L'espressione che segue *è in* ***nero corsivo***.

anche se si consiglia di usarle, se proprio necessarie, per porzioni di testo consistenti e non per singole parole come qui.

Modificare il corpo

Il corpo effettivo del font in un documento dipende da tre fattori:

- la classe di documento con cui si sta scrivendo;
- l'opzione (eventualmente) assegnata alla classe;
- le (eventuali) dichiarazioni per modificare il corpo del font date nel testo.

Le dichiarazioni elencate nella tabella 44 nella pagina seguente modificano il *corpo* del font. Naturalmente, anche il testo nella tabella risente dei fattori appena elencati: in particolare, si noti che il corpo di `\normalsize` è quello del testo principale di questa guida.

```
Lettere {\Large grandi} e
{\scriptsize piccole}.
```

Lettere grandi e piccole.

Si noti che queste dichiarazioni *modificano anche l'interlinea* del capoverso interessato, com'è giusto che sia, ma solo se esso termina entro il loro raggio d'azione. Nei due esempi seguenti, `\par` produce effetti differenti in base alla sua posizione nel testo.

```
{\large Socrate: «Platone
mentirà nella frase
seguente».}\par
```

Socrate: «Platone mentirà nella frase seguente».

Tabella 44: Dichiarazioni per modificare il corpo del font.

Dichiarazione	Risultato
`\tiny`	Esempio
`\scriptsize`	Esempio
`\footnotesize`	Esempio
`\small`	Esempio
`\normalsize`	Esempio
`\large`	Esempio
`\Large`	Esempio
`\LARGE`	Esempio
`\huge`	Esempio
`\Huge`	Esempio

```
{\large Platone: «Socrate ha
detto il vero nella frase
precedente».\par}
```

Platone: «Socrate ha detto il vero nella frase precedente».

Come si può osservare nel primo esempio, fuori dal gruppo `\par` non funziona più, con un risultato finale poco gradevole. La scrittura corretta pertanto è la seconda.

Questo viaggio tra stili e dimensioni si conclude con un simpatico consiglio, che starà all'utente seguire o meno:

> **Ricorda!** *Quanti* Più corpi e stili scegli di usare **in** un `documento`, *tanto* più LEGGIBILE e bello *diventa*.

Scegliere i font

A volte, per i motivi più vari, potrebbe essere necessario usare font diversi da quelli predefiniti: per una precisa richiesta dell'editore, per aumentare la leggibilità del documento, per "alleggerire" il proprio lavoro di qualche pagina caricando un font più compatto, eccetera.

LaTeX permette di usare praticamente *ogni tipo* di font in circolazione, ma si noti bene che un font *non* va usato per differenziare parti di testo (si usino per questo i comandi descritti nei paragrafi precedenti) né tanto meno per abbellirle. Analogamente, non si considerino i vari stili come una "tavolozza" da cui attingere a piacere: tondo, corsivo e maiuscoletto, da usare secondo i criteri spiegati nell'appendice A a pagina 149, dovrebbero bastare per la gran parte delle esigenze.

In linea generale, in un documento servono quattro famiglie di font:

- una famiglia con grazie per il testo principale;
- una famiglia senza grazie per scopi particolari;
- `una famiglia a larghezza fissa per gli indirizzi Internet, i codici e alcune parole di ambito informatico;`
- una famiglia per scrivere la matematica.

Scegliere famiglie che stiano bene insieme richiede abilità ed esperienza: perciò si consiglia di affidarsi senz'altro a classi e pacchetti (anche non standard, come classicthesis o arsclassica) che abbiano già risolto questo problema.

9.2.3 Lingue straniere

Se in un documento le parti in lingua straniera sono molte, anziché usare i comandi standard per le lingue descritti nel paragrafo 3.2 a pagina 18 è senz'altro più conveniente ridefinirli come si spiega di seguito.

Si può ottenere una versione personale di `\foreignlanguage` con

```
\newcommand{\inglese}[1]{\foreignlanguage{english}{\em #1}}
```

da usare come mostra l'esempio seguente:

```
Come scrisse una volta Donald
Knuth, <<\inglese{we have seen
that computer programming
is an art}>>.
```

Come scrisse una volta Donald Knuth, «*we have seen that computer programming is an art*».

Con le opportune sostituzioni si possono definire comandi per tutte le lingue di cui si ha bisogno.

In modo analogo si può ridefinire l'ambiente `otherlanguage*`

```
\newenvironment{inglese}%
   {\begin{otherlanguage*}{english}\em}%
   {\end{otherlanguage*}}
```

(o `otherlanguage`, se serve) da usare nel modo consueto.

Anche l'ambiente `quoting` presentato nel paragrafo 4.7.3 a pagina 55 potrà essere ridefinito per eventuali citazioni in lingua straniera. Si noti che in tutti questi esempi si è usata la dichiarazione `\em` per mettere in evidenza il testo straniero, secondo le consuetudini italiane.

9.3 SPECIALITÀ

> Il tutto è maggiore della somma delle parti.
>
> Aristotele
> *Metafisica*

In questo paragrafo si descrivono soltanto alcune delle numerose possibilità offerte da LaTeX per personalizzare il testo del proprio documento. Per ulteriori modifiche, si rimanda al paragrafo 9.5 a pagina 140.

9.3.1 Epigrafi

Esigenze *molto* particolari potrebbero richiedere di scrivere un'epigrafe in testa a un capitolo o un paragrafo: per realizzarla facilmente c'è il pacchetto epigraph. Un'epigrafe come quella che si vede dopo il titolo di questa sezione si ottiene con il seguente codice:

```
\section{Specialità}
\epigraph{Il tutto è maggiore della somma delle parti}%
         {Aristotele\\ \emph{Metafisica}}
```

9.3.2 Capilettera

Le convenzioni tipografiche permettono, a scopo decorativo, di sostituire la prima lettera di un capitolo con un *capolettera*, cioè una lettera di corpo maggiore delle altre (come qui). I capilettera hanno le forme più svariate: si va dalle lettere nel font corrente a caratteri elaboratissimi costruiti appositamente per pubblicazioni particolarmente ricercate.

Ne offre un vasto assortimento il pacchetto lettrine (se ne veda la documentazione), che per funzionare al meglio richiede a sua volta di caricare *prima* di fontenc con l'opzione T1 il pacchetto type1ec (per "scalare a piacimento" i font standard di LaTeX disegnati in vari corpi; usando font disegnati in un solo corpo come per esempio quelli dei pacchetti txfonts, pxfonts, mathpazo e fourier, invece, type1ec non serve):

```
\usepackage{type1ec}
\usepackage[T1]{fontenc}
\usepackage{lettrine}
```

Il comando `\lettrine`, la cui sintassi dovrebbe essere chiara, produce il capolettera:

```
\lettrine{⟨capolettera⟩}{⟨eventuale testo in maiuscoletto⟩}
```

Di seguito si riporta un esempio di testo con capolettera.

```
\lettrine{L}{e convenzioni}
tipografiche permettono, a scopo
decorativo, di sostituire la
prima lettera di un capitolo con
un \emph{capolettera}, cioè una
lettera di corpo maggiore delle
altre (come qui).
```

Le convenzioni tipografiche permettono, a scopo decorativo, di sostituire la prima lettera di un capitolo con un *capolettera*, cioè una lettera di corpo maggiore delle altre (come qui).

Una consuetudine molto seguita prevede di sfumare il passaggio dal capolettera al testo normale con qualche parola in maiuscoletto, come si è fatto in questo paragrafo: si può fare, ma *non* è una regola.

9.3.3 Scritture curiose

Con LaTeX, il pacchetto shapepar permette di "sagomare" un testo in molte forme diverse: cuori (come qui), quadrati, diamanti, cerchi, stelle a cinque punte, rettangoli, esagoni, candele che bruciano, triangoli diversamente orientati (si può anche ottenere un testo nella forma del logo TeX). Basta scrivere il testo da sagomare nell'argomento del comando corrispondente (`\heartpar` in questo caso, se ne veda la documentazione). Forse questa possibilità non verrà molto usata, ma è un chiaro esempio di come con pochissima fatica si possano ottenere risultati impensabili con altri programmi. Avete notato il piccolo cuore proprio qui sotto?

♡

9.3.4 Documenti multicolonna

Scrivere su più colonne

L'opzione di classe `twocolumn` permette di comporre l'intero documento su due colonne, ma ha il difetto di non bilanciare il testo nell'ultima pagina.

Il pacchetto multicol (se ne veda la documentazione) risolve il problema. Scrivendo semplicemente

```
\begin{multicols}{⟨numero⟩}
⟨testo multicolonna⟩
\end{multicols}
```

dove:

- ⟨*numero*⟩ è il numero delle colonne (da 2 a 10) in cui si intende dividere il testo;
- ⟨*testo multicolonna*⟩ si spiega da sé;

si ottiene un testo su più colonne, bilanciate automaticamente e sezionabili con i comandi consueti. Si noti che l'ambiente `multicols` può essere usato per un intero documento o solo per alcune sezioni di esso, e che non accetta note a margine né oggetti mobili che non siano a tutta pagina, ciò che lo rende adatto piuttosto a documenti di solo testo o a sezioni particolari come l'indice analitico.

Inserire oggetti mobili

L'inserimento di oggetti mobili in contesti multicolonna richiede un'attenzione particolare, pena risultati indesiderati. Le seguenti indicazioni dovrebbero risolvere i casi problematici.

Se il documento è composto su due colonne con `twocolumn`:

- si possono inserire oggetti nelle singole colonne mettendoli negli ambienti `table` e `figure` con una larghezza riferita a `\columnwidth` (che è la giustezza della colonna di composizione corrente);
- li si possono inserire a piena pagina mettendoli negli ambienti `table*` e `figure*` con una larghezza riferita a `\textwidth`, ma si noti che nel documento finito compariranno *sempre* in testa alla pagina (le eventuali preferenze indicate, infatti, vengono ignorate) o in una pagina `p`;
- si noti che la contemporanea presenza di entrambe le forme degli ambienti altera l'ordine di apparizione degli oggetti nel documento finito: risolve il problema il pacchetto dblfloatfix, che permette di indicare le preferenze `[tbp]`.

L'ambiente `multicols`, invece, non è indicato per contenere oggetti mobili. Infatti:

- `table` e `figure` non funzionano e al loro posto vanno usate le forme asteriscate, che si comportano come sopra;
- eventuali oggetti nelle singole colonne si possono solo collocare a mano e "proprio lì" con la preferenza `[H]` del pacchetto float (si veda il paragrafo 10.2 a pagina 144).

9.4 PERSONALIZZARE GLI OGGETTI MOBILI

9.4.1 Didascalie

Personalizzare la didascalia

Il pacchetto caption personalizza finemente le didascalie degli oggetti mobili, permettendo di regolarne *ogni* aspetto. Quelle di questa guida, per

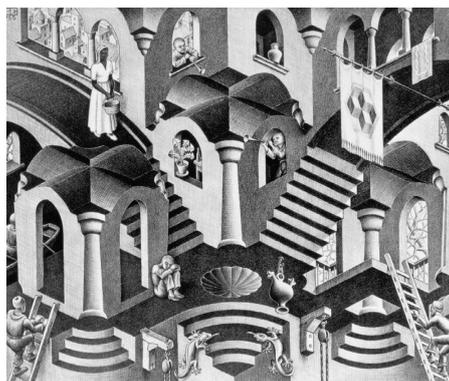



Figura 11: Esempio d'uso di sidecap (l'immagine riproduce la litografia *Concavo e convesso* di M. Escher).

esempio, sono state composte aggiungendo le seguenti opzioni a quelle già viste nel paragrafo 6.2.2 a pagina 88:

```
\captionsetup{format=hang,labelfont={sf,bf}}
```

dove:

- `format=hang` allinea (*hang*) alla prima riga quelle successive (LaTeX centra automaticamente le didascalie che occupano una sola riga);
- `labelfont={sf,bf}` imposta l'etichetta della didascalia in caratteri senza grazie e in nero.

Didascalie laterali: sidecap

Il pacchetto sidecap stampa una didascalia *accanto* all'oggetto (di solito una figura, più raramente una tabella) anziché sopra o sotto di esso. Le opzioni fondamentali del pacchetto sono le seguenti:

`outercaption` La didascalia è posta nel margine esterno della pagina (a sinistra in quelle pari e a destra in quelle dispari). È l'opzione predefinita.

`innercaption` La didascalia è posta nel margine interno della pagina (a destra in quelle pari e a sinistra in quelle dispari).

`leftcaption`, **`rightcaption`** La didascalia è sempre posta a sinistra o a destra dell'oggetto, rispettivamente.

`ragged`, **`raggedright`**, **`raggedleft`** Gestiscono l'allineamento delle didascalie brevi.

Il pacchetto definisce due nuovi ambienti `SCfigure` e `SCtable` (analoghi agli ambienti `figure` e `table`) che prevedono due argomenti facoltativi,

```
\begin{SCfigure}[⟨larghezza relativa⟩][⟨collocazione⟩]
```

e

```
\begin{SCtable}[⟨larghezza relativa⟩][⟨collocazione⟩]
```

dove:

- ⟨*larghezza relativa*⟩ indica il rapporto fra larghezza della didascalia e larghezza della figura (o della tabella) corrispondente. Un valore elevato di ⟨*larghezza relativa*⟩ (`50`, per esempio) imposta per la didascalia la larghezza massima possibile. Il valore predefinito è `1`.
- ⟨*collocazione*⟩ indica le preferenze di collocazione degli ambienti mobili, da usare nel modo consueto. Il valore predefinito è `tbp`.

Esistono anche gli ambienti SCfigure* e SCtable* (la cui sintassi è la stessa di SCfigure e SCtable) che in un documento a più colonne permettono di ottenere rispettivamente una figura o una tabella mobile (con didascalia laterale) estesa su tutta la pagina.

La figura 11 nella pagina precedente è stata inclusa nel documento con un codice del tipo:

```
\begin{SCfigure}[]
\centering
\includegraphics[width=0.5\textwidth]{ConcavoConvesso}
\caption{Un esempio di figura con didascalia laterale.}
\label{fig:sidecap}
\end{SCfigure}
```

9.4.2 Nuove disposizioni

Oggetti immersi nel testo: wrapfig

Ci sono circostanze in cui può essere desiderabile "avvolgere" un oggetto con del testo, magari anche solo per movimentare la pagina. Risolve il problema il pacchetto wrapfig, particolarmente indicato perché interagisce correttamente con caption per personalizzare opportunamente la didascalia.

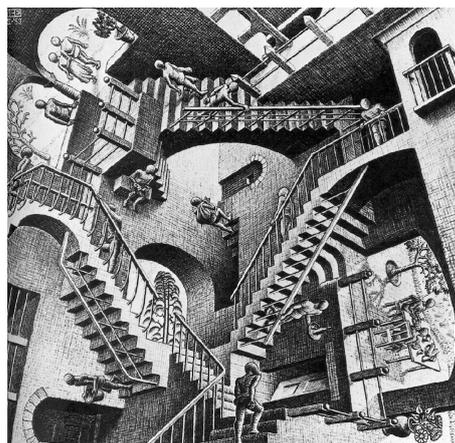



Figura 12: Esempio d'uso di wrapfig (l'immagine riproduce la litografia *Relatività* di M. Escher).

Il pacchetto definisce l'apposito ambiente wrapfloat, nel quale mettere l'oggetto con i comandi consueti. Ragioni estetiche impongono di circondarlo soltanto con testo "puro" e di rimandare più oltre l'eventuale inserimento di altri oggetti. Tuttavia, anche operando correttamente il pacchetto non garantisce un risultato ottimale già alla prima composizione: potrebbero essere necessari numerosi aggiustamenti manuali della pagina, più o meno consistenti.

Come si può osservare, una wrapfig correttamente ottenuta dà un risultato altamente professionale, ma si raccomanda di sfruttare questa possibilità soltanto in circostanze davvero eccezionali. Le condizioni che garantiscono la buona riuscita dell'operazione, infatti, sono particolari: la composizione deve essere a piena pagina e la colonna di testo sufficientemente grande da poter comprendere una figura larga la sua metà: in caso contrario, la colonna di testo accanto all'immagine risulterà troppo stretta e inevitabilmente solcata da ruscelli. La regola da seguire rimane la stessa: includere tutti gli oggetti fuori testo e lasciar fare a LaTeX. Soltanto a lavoro ultimato, quando il risultato finale è davvero soddisfacente, sarà possibile ricollocare qualcuno degli oggetti secondo il proprio gusto.

La figura 12 è stata ottenuta con un codice del tipo:

```
\dots ''Avvolgere'' un oggetto con del testo è molto semplice.

\begin{wrapfloat}{figure}{I}{0pt}
\includegraphics[width=0.5\textwidth]{Relativo}
\caption{Esempio di figura ''avvolta'' da un testo.}
```

Tabella 45: Preferenze di collocazione dell'ambiente `wrapfloat`.

`r`, `R`	Sul lato destro del testo (*right*)
`l`, `L`	Sul lato sinistro del testo (*left*)
`i`, `I`	Sul margine interno (*inner*)
`o`, `O`	Sul margine esterno (*outer*)

```
\end{wrapfloat}

Può rendersi necessario un po' di lavoro per ottenere
un'impaginazione ottimale. \dots
```

Come si vede nell'esempio precedente, l'ambiente `wrapfloat` richiede tre argomenti obbligatori:

```
\begin{wrapfloat}{⟨oggetto⟩}{⟨collocazione⟩}{⟨larghezza⟩}
```

Dove:

- ⟨*oggetto*⟩ indica il tipo di *oggetto* da includere (`figure` o `table`, da non confondersi con gli ambienti omonimi);
- ⟨*collocazione*⟩, che dice a LaTeX dove mettere l'oggetto sulla pagina, accetta *una sola* delle otto opzioni della tabella 45, in maiuscolo o in minuscolo a seconda che si voglia stampare l'oggetto "esattamente qui nel testo" o che si voglia creare un oggetto mobile, rispettivamente;
- ⟨*larghezza*⟩ specifica la larghezza dell'oggetto che, se nulla (`0pt`), equivale all'opzione assegnata al comando `\includegraphics`.

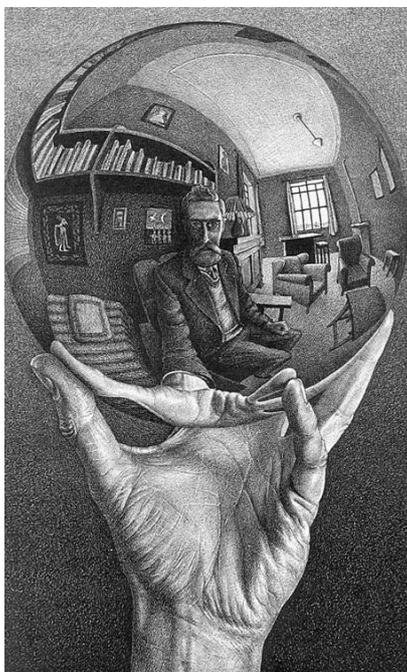
Oggetti multipli: subfig

Il pacchetto subfig (che richiede caption) gestisce con `\subfloat` più (sotto)figure (come la figura 13 a fronte) o (sotto)tabelle in un unico ambiente mobile e ne regola le didascalie molto finemente.

La figura 13 nella pagina successiva è stata ottenuta con un codice del tipo:

```
\begin{figure}
\centering
\subfloat[][\emph{Mano con sfera riflettente}.]
   {\includegraphics[width=.45\textwidth]{Sfera}} \quad
\subfloat[][\emph{Belvedere}.]
   {\includegraphics[width=.45\textwidth]{Belvedere}} \\
\subfloat[][\emph{Cascata}.]
   {\includegraphics[width=.45\textwidth]{Cascata}} \quad
\subfloat[][\emph{Salita e discesa}.]
   {\includegraphics[width=.45\textwidth]{SalitaDiscesa}}
\caption{Alcune litografie di M.~Escher.}
\label{fig:subfig}
\end{figure}
```

Si noti che:

- nel primo argomento facoltativo di `\subfloat`, se usato, si mette la didascalia breve da mandare nel relativo indice (`\listoffigures` o `\listoftables`);
- nel secondo va messa la didascalia che comparirà effettivamente sulla pagina;

(a) *Mano con sfera riflettente.*

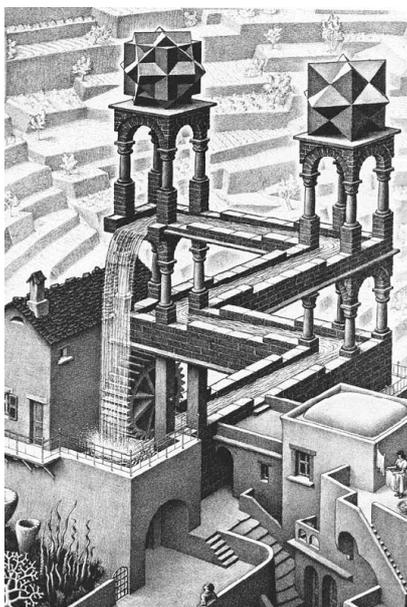
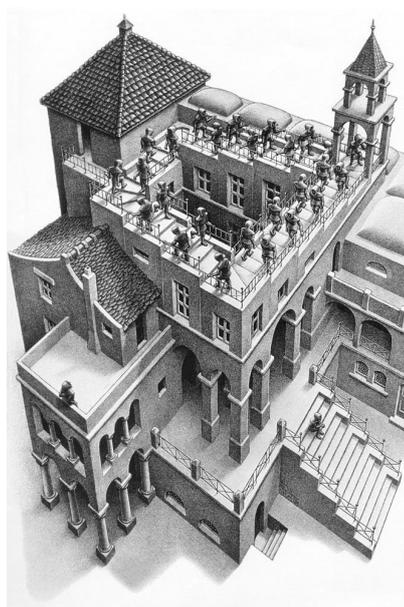
(b) *Belvedere.*

(c) *Cascata.*

(d) *Salita e discesa.*

Figura 13: Esempio d'uso di subfig (le immagini riproducono alcune litografie di M. Escher).

Tabella 46: Voci italiane di babel.

Comando	Voce	Comando	Voce
`\abstractname`	Sommario	`\indexname`	Indice analitico
`\alsoname`	vedi anche	`\listfigurename`	Elenco delle figure
`\appendixname`	Appendice	`\listtablename`	Elenco delle tabelle
`\bibname`	Bibliografia	`\pagename`	Pag.
`\ccname`	e p. c.	`\partname`	Parte
`\chaptername`	Capitolo	`\prefacename`	Prefazione
`\contentsname`	Indice	`\proofname`	Dimostrazione
`\enclname`	Allegati	`\refname`	Riferimenti bibliografici
`\figurename`	Figura	`\seename`	vedi
`\glossaryname`	Glossario	`\tablename`	Tabella
`\headtoname`	Per		

- per riferirsi a un sottooggetto in particolare da altre parti del documento, `\label` va dato *dentro* il secondo argomento facoltativo immediatamente dopo la sottodidascalia;
- il comando matematico `\quad` agisce anche in modo testuale.

9.5 ALTRE PERSONALIZZAZIONI

9.5.1 Modificare le parole fisse

Per modificare le parole fisse generate da babel (sostituire *Capitolo* con *Unità* oppure *Bibliografia* con *Letture suggerite*, per esempio), lo si carica e si scrive *nel preambolo* il codice

```
\addto\captions⟨lingua⟩{⟨testo⟩}
```

che comanda a LaTeX di aggiungere alle definizioni specifiche della ⟨*lingua*⟩ il ⟨*testo*⟩.

Volendo sostituire la voce *Capitolo* con *Unità* basta scrivere

```
\addto\captionsitalian{\renewcommand{\chaptername}{Unità}}
```

mentre per sostituire *Bibliografia* con *Letture suggerite* si può scrivere

```
\addto\captionsitalian{\renewcommand{\bibname}{Letture suggerite}}
```

La tabella 46 elenca i comandi di queste voci con la relativa traduzione italiana.

9.5.2 Personalizzazioni avanzate

Di seguito si descrivono brevemente alcuni tra i pacchetti più usati per personalizzare in modo avanzato certi aspetti del documento. Si raccomanda di usarli, se proprio necessario, quando il proprio lavoro è in forma *definitiva* e dopo averne letto attentamente la documentazione o la descrizione contenuta in [Gregorio, 2010].

titlesec, titletoc Permettono di personalizzare ogni aspetto dei titoli di sezione, dell'indice generale, delle testatine e dei piedi.

sectsty Permette di cambiare il font dei titoli di sezione e la loro posizione sulla pagina.

tocloft Permette di cambiare la resa tipografica degli indici (generale, delle tabelle e delle figure) e di definirne di nuovi.

fancyhdr Permette di personalizzare ogni aspetto di testatine e piedi.

enumitem Permette di personalizzare ogni aspetto dei tre ambienti standard per gli elenchi `itemize`, `enumerate` e `description`.

xcolor Permette di gestire i colori in un documento.

10 REVISIONE FINALE

Come s'è detto più volte, anche se durante la stesura LaTeX induce l'autore a privilegiare la struttura logica del documento e a trascurarne l'aspetto finale, nemmeno lui, tuttavia, riesce a risolvere in una volta sola tutti i problemi tipografici risparmiando interventi in prima persona sulla forma del lavoro: è il piccolo prezzo da pagare per un prodotto di altissima qualità. La revisione finale di un documento precedente la stampa è una fase delicata: si tratta, a volte, di risolvere grandi problemi d'impaginazione con piccoli (o piccolissimi) aggiustamenti. Un'arte complicata ma ricca di soddisfazioni, che dà i propri frutti migliori solo quando il documento è nella sua forma *definitiva*.

Prima di attuare i suggerimenti proposti nei prossimi paragrafi si tenga presente che *quasi* sempre si può correggere un difetto d'impaginazione semplicemente riformulando il testo nel punto o nei dintorni del punto critico. Si risolva un problema alla volta, un capoverso alla volta: *probabilmente* andrà a posto da solo anche qualche altro difetto più in là nel testo.

10.1 PROBLEMI ORIZZONTALI

I difetti d'impaginazione orizzontali riguardano la formazione dei capoversi, e di solito prendono la forma di righe che sporgono nel margine destro della pagina.

Titoli problematici

Quando sono troppo lunghi, i titoli creano problemi in almeno due luoghi:

- nel corpo del documento, perché LaTeX potrebbe spezzarli in modo non soddisfacente o addirittura farli sporgere se non trova un punto di sillabazione adatto;
- nell'indice generale (e nelle eventuali testatine), riproducendo il problema.

Un titolo di sezione non dovrebbe *mai* andare a capo. Si risolvono tutti i problemi in una volta sola riformulandolo: si può fare praticamente sempre. Se invece il titolo lungo fosse davvero necessario, lo si mandi a capo con un \\ esplicito nel corpo del documento e si usi l'argomento facoltativo del comando di sezionamento per l'indice generale e le testatine:

```
\chapter[Titolo breve per indice e testatine]%
       {Titolo lungo\\ da mandare a capo}
```

Capoversi problematici

Un capoverso che LaTeX non riesce a comporre bene può influenzare la divisione in pagine del documento, per cui sistemarlo potrebbe risolvere

automaticamente anche alcuni dei problemi descritti nel prossimo paragrafo. Di seguito si danno alcuni consigli per le situazioni più frequenti.

Un documento di una certa lunghezza conterrà quasi certamente qualche riga che LaTeX non è riuscito a comporre "bene" e che si vedrà sporgere (ma non sempre, si noti) nel margine destro. Per essere davvero sicuri di "curarle" tutte, si possono evidenziare dando immediatamente prima dell'inizio del documento il comando

```
\overfullrule=5pt
```

che stampa un ▮ accanto a ciascuna (si può variare la larghezza del rettangolino aumentando il valore in punti).

Possono causare lo stesso problema anche le parole che LaTeX per vari motivi non riesce a dividere e che dopo la composizione si trovano a fine riga: si può tentare di suggerirne la sillabazione con degli `\-` espliciti là dove serve.

Si possono risolvere altre situazioni problematiche eliminando qualche spazio indivisibile non necessario (*mai*, però, tra un'iniziale puntata e il relativo cognome) o mettendo in display alcune formule matematiche che in linea sono poco leggibili. Infine, si usino il meno possibile lunghi URL nel corpo del testo: li si metta in una nota al piede (se sono pochi ed è proprio necessario) oppure, se sono molti, in un elenco alla fine del documento.

Gli ultimi tocchi al proprio documento provengono da considerazioni di carattere estetico. Si eviti che due righe consecutive comincino o finiscano con la stessa o le stesse parole riformulando il testo; si eviti che un capoverso termini con una sola parola molto breve (o, peggio, con una sola sillaba) racchiudendola in `\mbox` e unendola alla penultima con uno spazio indivisibile. Lo spazio sottile `\,` è utilissimo per risolvere altri inestetismi: può servire per separare un apostrofo e le successive virgolette inglesi; per evitare fastidiose sovrapposizioni in certe sequenze: tra una lettera corsiva con tratti discendenti come la f e una parentesi tonda immediatamente prima, per esempio, oppure tra l'esponente di una nota al piede e ciò che lo precede.

10.2 PROBLEMI VERTICALI

I difetti d'impaginazione verticali riguardano la divisione in pagine del documento.

Oggetti in testo e fuori testo

Gli oggetti in testo causano problemi d'impaginazione a volte senza soluzione perché, di regola, non possono essere spezzati *in nessun modo* su più pagine. Basta evitarli il più possibile e non si avranno di questi problemi.

Gli oggetti fuori testo, invece, sono molto più flessibili. Tenendo presente che fintanto che oggetto e relativo riferimento si trovano sulla stessa pagina o al massimo in due pagine opposte il risultato si può considerare ottimale (a libro aperto non dovrebbe essere necessario voltare pagina, per intenderci), anche se purtroppo non sempre è possibile, nel documento composto qualche oggetto potrebbe non piacere dove LaTeX ha pensato di metterlo:

- potrebbe essere finito un po' troppo lontano dal punto "ottimale";
- potrebbe essere finito in una pagina insieme a troppi titoli di sezione, ciò che renderebbe la pagina troppo vuota;
- ci si potrebbe trovare con due oggetti in una pagina e nessuno in quella successiva, quando li si preferirebbe più omogeneamente distribuiti;

- si potrebbe volere in una pagina di soli oggetti un oggetto che invece LaTeX ha messo in una pagina con del testo.

Come fare?

I primi tre casi si risolvono semplicemente facendo arretrare il codice dell'oggetto di qualche capoverso, mentre il terzo si risolve dando all'oggetto la preferenza `[p]`, che LaTeX riesce a soddisfare *sempre* anche se da sola. Tutte le altre situazioni intricate si possono sbrogliare con un uso oculato delle preferenze di collocazione e qualche prova. Si ottengono i risultati migliori se il testo è molto e gli oggetti pochi, mentre si potrebbero avere problemi nel caso contrario.

Si danno, infine, ulteriori suggerimenti per trarsi d'impaccio:

- a volte può risolvere la situazione `\clearpage`, spiegato nel paragrafo 10.2 nella pagina successiva;
- si può provare con `\FloatBarrier` (richiede il pacchetto placeins), che traccia delle barriere invalicabili per gli oggetti mobili;
- in casi estremi, e solo per ottenere effetti particolari, si può forzare l'oggetto nella posizione desiderata con la preferenza `[H]` del pacchetto float (da usare *sempre* da sola).

```
\dots
qui finisce un capoverso.

\begin{figure}[H]
\centering
\includegraphics[width=%
0.5\textwidth]{Formica}
\caption{Figura collocata
a mano.}
\label{fig:float}
\end{figure}

La figura~\vref{fig:float} è
un esempio di figura mobile
collocata a mano.
```

…qui finisce un capoverso.

Figura 14: Figura collocata a mano.

La figura 14 è un esempio di figura mobile collocata a mano.

Note a piè di pagina

Le note al piede sono problematiche specie se lunghe più di un capoverso, perché sono ottime candidate a finire tra due pagine moltiplicando i difetti. Le si eviti per quanto possibile, dunque.

Orfani e vedove

In tipografia si usa chiamare *orfano* la prima e unica riga di un capoverso in fondo alla pagina e *vedova* l'ultima riga di un capoverso in cima a una pagina nuova. Che siano gli orrori tipografici da evitare con più cura è testimoniato dalla terminologia tedesca: una riga orfana si chiama *Schusterjunge* ("apprendista ciabattino"), mentre una riga vedova *Hurenkind* ("figlio di p***"). Bisogna fare in modo che alle due estremità di una pagina ci siano sempre *almeno due* righe di uno stesso capoverso, e almeno quattro nell'ultima pagina di un capitolo. È accettabile, invece, una riga solitaria in cima alla pagina immediatamente seguita da un ambiente in display di qualunque tipo. LaTeX è programmato per evitare automaticamente queste due situazioni: quando non ce la fa, si ricorra ai consigli descritti di seguito.

Il metodo più semplice per trarsi d'impaccio è riformulare il capoverso in questione per diminuirlo o aumentarlo di una riga (in entrambi i casi si risolve il problema) oppure valutare la possibilità di spezzare un capoverso in due o di riunirne due in uno nei pressi del problema.

Se il capoverso che contiene l'orfano o la vedova è abbastanza lungo, si può provare a dare *immediatamente dopo* il punto di fine capoverso uno dei due comandi

```
\looseness=1
\looseness=-1
```

con cui si chiede a LaTeX di provare ad allungare o accorciare il capoverso del numero di righe indicato (che è `0` per impostazione predefinita). Un tentativo con `-1` potrebbe funzionare se l'ultima riga è *molto* breve, uno con `1` se è quasi piena. Si ricordi che il comando agisce soltanto sul capoverso cui è applicato.

Si possono eliminare orfani e vedove anche spostando una riga di testo dalla pagina *precedente* a quella in cui si trovano con `\pagebreak`. Il comando va dato prima della riga orfana nella versione composta e funziona tanto meglio quanto più la pagina contiene spazi flessibili. Lo si usi senz'altro quando un display sta in cima alla pagina e il capoverso precedente è lungo una o due righe soltanto. Si ricorda anche l'opzione `noorphans` dell'ambiente `quoting`.

Se nessuno degli strumenti visti sin qui funziona, si può provare con uno tra

```
\enlargethispage{1\baselineskip}
\enlargethispage{-1\baselineskip}
```

che dicono a LaTeX di allungare o accorciare la pagina del numero di righe indicato nell'argomento: è bene che l'allungamento (o l'accorciamento) ammonti a una riga soltanto o, *in casi particolarissimi*, a due, ma non di più. Il comando *modifica la gabbia del testo* e va dato *tra due capoversi*: perciò *va usato in coppia su due pagine opposte*, il cui piede non sia occupato da note o numeri di pagina e con margini inferiori sufficientemente alti. Se per la stampa si userà una carta di qualità, infine, l'aggiustamento non si noterà nemmeno. La variante asterisco del comando si comporta allo stesso modo, provando prima a comprimere o ad allargare gli eventuali spazi bianchi sulla pagina a seconda dei casi.

Si ricordi, infine, che salvo situazioni fortunate i capoversi brevi non sono adatti per simili acrobazie.

Interrompere la pagina corrente

Di seguito si descrivono brevemente i principali comandi standard per "cambiare pagina". Tutti interrompono la pagina corrente *nel punto in cui vengono dati*, ma:

- `\newpage` comincia semplicemente una pagina nuova;
- `\pagebreak` prima di cominciare la nuova pagina stiracchia in verticale il contenuto di quella in cui viene dato per riempirla al meglio;
- `\clearpage` prima di cominciare la nuova pagina dice a LaTeX di stampare tutti gli oggetti già definiti e che non hanno ancora trovato posto sulle pagine, se ce ne sono (si veda il paragrafo 6.2.2 a pagina 87);
- `\cleardoublepage` si comporta come il comando precedente ma inserendo, se necessario, una pagina bianca prima di cominciare quella nuova (utile nei documenti impostati per la stampa in fronte/retro).

Si tenga *ben* presente che è LaTeX a interrompere automaticamente la pagina nel punto per lui migliore: i comandi appena descritti ne modificano il comportamento predefinito e possono risolvere situazioni problematiche, ma vanno usati con grande accortezza.

Controllare l'indice analitico

Per controllare l'indice analitico può essere utile il pacchetto showidx, che ne visualizza le voci nel margine sinistro della pagina.

E per finire...

Infine, non rimane che rileggere il tutto più volte per stanare i refusi: è praticamente impossibile non scovarne in ogni documento che superi la decina di pagine.

A NORME TIPOGRAFICHE

Quest'appendice, basata su [Cevolani, 2006] e [Lesina, 2004], cui si rimanda per ogni approfondimento, descrive sinteticamente e senza pretese di completezza le tradizioni tipografiche più seguite nella redazione di un documento in italiano. Di ogni regola discussa si mostra, per quanto possibile, l'applicazione in LaTeX. La parola *norma* va qui intesa in senso piuttosto lato: anche nella nostra lingua, come in tutte le altre, non esistono che pochissime regole tipografiche realmente universali e vincolanti, mentre molti aspetti del testo finito dipendono da tradizioni e abitudini o dal gusto dell'autore o dell'editore del testo.

Ciò premesso, si tenga presente che la scrittura in una qualunque lingua è governata nel mondo dalle norme emesse dall'ISO e in Italia da quelle definite dall'UNI, che hanno valore di legge. Per maggiori dettagli su queste norme, se ne veda la documentazione.

Naturalmente non serve ricordare che per risolvere dubbi di altro tipo il mezzo più rapido e sicuro rimane la consultazione di grammatica e dizionario.

A.1 ACCENTO E APOSTROFO

Accento

In italiano ci sono due accenti:

- *grave*, `, che indica suono aperto;
- *acuto*, ´, che indica suono chiuso.

Un terzo accento, il *circonflesso*, ˆ, usato in passato soprattutto per distinguere gli omografi uscenti in *-ii* al plurale — *odî* (plurale di *odio*) da *odi* (voce del verbo *udire*) — oggi non viene quasi più usato.

La loro applicazione non è univoca, e oscilla tra le prescrizioni UNI (per cui l'accento grave può colpire *tutte* le vocali: à/À, è/È, ì/Ì, ò/Ò, ù/Ù; e l'accento acuto soltanto le vocali *e* e *o*: é/É, ó/Ó) e le tradizioni di alcune case editrici (per cui *i* e *u* portano sempre l'accento acuto, per esempio). Qualunque sistema si scelga, l'importante è seguirlo coerentemente in *tutto* il documento.

Come si può osservare, gli accenti vanno apposti *sopra* la vocale, minuscola o maiuscola che sia: i pacchetti fontenc e inputenc opportunamente impostati interpretano e stampano correttamente le lettere accentate direttamente dalla tastiera (si veda il paragrafo 3.2 a pagina 18). Si noti tuttavia che per ottenere alcuni caratteri assenti dalla tastiera italiana come la *ó* è sempre necessario digitare il codice esplicito o le combinazioni di tasti dedicate. Al contrario di quanto accade con le vecchie macchine per scrivere o con altri elaboratori di testo, LaTeX permette di accentare senza problemi *anche le lettere maiuscole*, che spesso vengono erroneamente apostrofate (come in *E'*, che sta per "ei" cioè "egli").

Apostrofo

L'*apostrofo*, ', segnala normalmente la caduta della parte iniziale, come in *'sta* (per *questa*), o finale di una parola, come in *un'altra*, *un po'*, *da'* (imperativo), eccetera.

Quando cadono lettere o sillabe iniziali l'apostrofo è preceduto da uno spazio; quando cadono sillabe finali l'apostrofo è seguito da uno spazio o da un segno di interpunzione.

In una parola in cui cade una vocale finale di fronte a un'altra parola che inizia per vocale (come in *un'altra* o in *quell'uomo*), l'apostrofo sta al posto della vocale caduta e la parola seguente comincia senza nessuno spazio intermedio. Se `babel` è caricato con l'opzione `italian` LaTeX evita automaticamente l'apostrofo in fine di riga: *quell'uomo* viene sillabato "quel-l'uo-mo", come fosse un'unica parola (con altre lingue che usano l'apostrofo, però, questo potrebbe non succedere).

A.2 PUNTEGGIATURA E SPAZIATURA

La punteggiatura italiana comprende segni di interpunzione, parentesi, virgolette, puntini di sospensione, trattini e altri simboli come asterisco e barra. Ci sono alcune regole fisse sull'uso degli spazi prima e dopo i segni di interpunzione.

A.2.1 Segni d'interpunzione

Tutti i segni di interpunzione seguono immediatamente la parola che li precede e vanno separati con uno spazio da quella che li segue.

A.2.2 Virgolette

Nel testo non si lasciano spazi bianchi tra le virgolette e il loro contenuto, mentre li si lascia tra le virgolette e il testo esterno.

Normalmente si contrassegnano tra virgolette:

- citazioni dirette («Gli ho sentito dire: "Verrò di sicuro"») e citazioni "annidate", cioè citazioni dentro a citazioni;
- termini ed espressioni che specificano il significato di altre parole: «La parola *box* significa "scatola"»;
- espressioni figurate o gergali: «Si prevedono scioperi "a singhiozzo"»;
- termini correnti a cui si attribuisce un significato diverso da quello abituale: «Questo ragazzo non è certo una "cima"»;
- parole usate in senso ironico.

I casi sopra elencati non costituiscono regole realmente vincolanti: alcuni di essi, infatti, trovano un ottimo concorrente nel *corsivo*, con l'avvertenza di evitare *sempre* l'uso contemporaneo dei due metodi. Non esistono nemmeno regole fisse o comunemente accettate per l'uso dei vari tipi di virgolette (si veda il paragrafo 4.3.1 a pagina 48): nella pubblicazione dei testi a stampa, infatti, si adottano convenzioni tipografiche variabili dall'uno all'altro editore e perfino tra l'una e l'altra collana dello stesso marchio editoriale. L'unico consiglio che qui si può dare è di scegliere un sistema di virgolettatura *prima* di scrivere il documento e di usarlo poi con coerenza al suo interno.

A.2.3 Parentesi

In italiano si usano di solito le parentesi *tonde* ( ) e le parentesi *quadre* [ ]. Le parentesi *graffe* { } e *angolate* ⟨ ⟩ (i comandi per queste ultime sono `\langle` e `\rangle`, da usare in modo matematico) si trovano solo in discipline scientifiche, rispettivamente in matematica e in informatica.

Generalmente le parentesi racchiudono un *inciso* nel discorso (cioè una parte *che si può omettere*, come questa). Seguono alcune indicazioni generali sul loro impiego.

- Nel testo non si lasciano spazi bianchi tra le parentesi e il loro contenuto, mentre lo si lascia tra le parentesi e il testo esterno (a meno che non siano a fine enunciato, come qui).
- Quando un segno d'interpunzione ricorre dentro le parentesi, l'enunciato finirà, come questo, con un punto *fuori* dalla parentesi stessa (ovviamente!). Si noti che di norma solo i punti interrogativo ed esclamativo possono stare dentro le parentesi.
- L'uso di racchiudere tra parentesi un intero enunciato è lecito ma poco diffuso in italiano, a differenza di altre lingue. (In tal caso, anche la punteggiatura rientra nelle parentesi, come qui.)

Le parentesi quadre vengono usate quasi esclusivamente in due casi.

- Come parentesi interne a parentesi (come [anche se non è molto frequente] in questo caso).
- Per introdurre «il commento di una persona diversa dall'autore del testo cui il commento si riferisce [in questo modo]». In quest'ultimo caso rientra anche quello dell'*omissione* volontaria (che è comunque un commento), segnalata con [...] (si veda il paragrafo 4.3.1 a pagina 48).

A.2.4 Puntini di sospensione

I puntini di sospensione sono *sempre e solo tre* e, come gli altri segni di interpunzione, seguono immediatamente la parola che li precede e sono separati con uno spazio da quella che li segue... in questo modo. In LaTeX, il comando da usare è `\dots` (non vanno *mai* inseriti a mano tre punti separati), eventualmente inserendo uno spazio esplicito se i puntini non sono a fine enunciato (si veda il paragrafo 4.3.1 a pagina 48).

Se usati per indicare un'omissione in una citazione «è bene [...] inserirli entro parentesi quadre o tonde» con il comando `\omissis` come in questo caso (si veda lo stesso paragrafo).

A.2.5 Trattino, tratto e lineetta

Si è già spiegato (nel nel paragrafo 4.3.1 a pagina 48) come realizzare questi tre segni con LaTeX. Di seguito se ne descrive l'uso in termini generali.

Il *trattino* di solito divide le parti di un'espressione composta e nella scrittura s'interpone tra di esse senza ulteriori spazi bianchi. Si usa:

- Nei termini formati da due parole autonome, come «guerra-lampo».
- Negli intervalli numerici o di tempo i cui estremi siano espressi in cifre, come «1915-1918» e «pagine 2-11». Se gli estremi sono costituiti da più d'una parola, per maggiore chiarezza è opportuno spaziare anche il trattino da entrambe le parti: «Trentino - Alto Adige».

Il *tratto*, preceduto e separato da uno spazio bianco, di solito suddivide elementi come titoli, didascalie e diciture di vario tipo, come «Varianti di carattere – Il maiuscoletto».

La *lineetta* isola nettamente un inciso all'interno del testo o, usata in coppia, segnala le battute di dialogo nei testi narrativi. Nella scrittura del testo è separata dal testo precedente e seguente da uno spazio bianco normale o sottile — come in questo caso.

A.2.6 Barra e asterisco

La *barra* (o *sbarretta*) può essere usata nelle seguenti circostanze ordinarie.

- Senza spazi né prima né dopo, per un'alternativa tra due termini: «i passeggeri diretti a Torino/Milano».
- Preceduta e seguita da uno spazio bianco, separa i versi d'una poesia riportata in linea: «M'illumino / d'immenso».
- Senza spazi prima e dopo, si usa nelle frazioni numeriche in modo testuale: «i 3/4 della popolazione».

Si ricordi che la barra (*slash*, /), inseribile direttamente dalla tastiera, *non* va confusa con il carattere di barra rovescia (`\textbackslash`, \ ), riservato ai comandi di LaTeX.

L'*asterisco* ha un uso limitato alle seguenti circostanze:

- separato da quanto precede con uno spazio sottile e ripetuto tre volte, indica omissione volontaria («Il padre Cristoforo da * * * era un uomo più vicino ai sessanta che ai cinquanta», codice: `*\,*\,*`);
- per indicare in linguistica forme non attestate, scorrette o inaccettabili: « * che io vadi».

A.3 STILE DEL FONT

Oltre a quello normale usato per il testo, di solito un font prevede lo stile *corsivo*, **nero** e MAIUSCOLETTO. È poco utilizzato invece lo stile *inclinato* (da non confondere con il corsivo *inclinato*). Altri stili possono essere usati per esigenze particolari: per esempio, in questo documento si usa lo stile `dattilografico` per evidenziare il codice LaTeX. Lo stile "evidenziato" (o "enfatizzato") è reso normalmente con il corsivo, ma riveste un ruolo logico differente.

A.3.1 Corsivo

Convenzionalmente si scrivono in corsivo le seguenti categorie di parole:

- termini tecnici e specialistici: «Una *distribuzione* è una raccolta di programmi. . . »;
- parole o frasi straniere di uso non comune con le quali si ritiene che il lettore non abbia affinità: «Questa tecnica di *engraving* può essere considerata. . . »;
- parole e lettere a cui ci si riferisce come tali nel testo: «La lettera *e* non compare nella parola *parola*»;
- parole o frasi da evidenziare: «Questo *non* si fa».

Oltre che nel testo principale, si può usare il corsivo per comporre i titoli.

A.3.2 Nero

Nelle classi standard di LaTeX il nero si usa per i titoli dei capitoli, dei paragrafi e delle altre suddivisioni del testo. Si tende di norma a non impiegarlo per evidenziare parole o addirittura interi capoversi nel testo principale (il corsivo va benissimo), e comunque a utilizzarlo con moderazione per non appesantire la pagina.

A.3.3 Maiuscoletto

Il maiuscoletto viene usato in italiano quasi esclusivamente per i nomi degli autori citati in bibliografia, come in Bringhurst (1992). Questa convenzione, tuttavia, dipende dallo stile bibliografico scelto. Oppure concorre con il maiuscolo per segnalare gli acronimi nel testo.

A.4 TRATTAMENTO DEL TESTO

Per una trattazione completa di tutte le tipologie di parole si rimanda a [Lesina, 2004]. A titolo d'esempio, di seguito si prende in considerazione uno dei casi più frequenti, quello delle parole straniere.

A.4.1 Parole straniere

Le parole straniere vanno in corsivo, a meno che non vengano esplicitamente "quoted" ("virgolettate", come in questo caso) o non siano entrate nell'uso comune. Quindi si scriverà: «ho visto un bel film» (comune) ma «ho mangiato un *pudding*» (non comune). In realtà, dato che è molto difficile stabilire che cosa sia o meno "comune", la cosa più corretta da fare è scrivere in corsivo le parole straniere che si presumono di uso non corrente *per il lettore a cui ci si rivolge*. In un libro d'informatica, perciò, *software* e *computer* potranno essere scritte in tondo.

I nomi propri e le denominazioni ufficiali (come Stanford University e Magna Charta) non sono considerati parole straniere e vanno in tondo.

La traduzione straniera di un'espressione italiana usata nel testo può essere semplicemente messa in corsivo e fra parentesi (*bracket*), come in questo caso. Se invece l'espressione tradotta ricorre in una citazione e se ne vuole indicare la forma originale, occorre inserirla fra parentesi quadre come ogni altro commento. Per esempio: «La visione del mondo [*Weltanschauung*]...».

Ovviamente ogni lingua straniera va sillabata a sé, ed è altrettanto ovvio che scrivendo in lingue diverse dall'italiano alcune cesure nel documento finito potrebbero risultare errate. Si noti che è ammesso sillabare secondo le nostre regole parole straniere isolate nel discorso, ma non un testo in lingua più esteso (si veda il paragrafo 3.2 a pagina 18).

A.4.2 Numeri

Scrivere i numeri

I caratteri numerici possono essere *maiuscoli* e *minuscoli*. Si osservi la resa tipografica del numero *1821* nei due esempi seguenti:

```
Il numero $1821$ non è primo. \\
Napoleone morì nel 1821.
```

Il numero 1821 non è primo.
Napoleone morì nel 1821.

Nel primo, 1821 è maiuscolo; nel secondo, 1821 è minuscolo.

I numeri maiuscoli presentano tutti la stessa altezza e sono indicati:

- per scrivere le formule matematiche;
- nelle tabelle composte di dati numerici;
- quando li si intende in senso aritmetico (nell'indicazione di quantità esatte, per esempio).

Mentre i primi due punti si risolvono con gli strumenti già esaminati nei capitoli precedenti, per l'ultimo c'è più di una possibilità: o si scrivono i numeri tra i comandi per le formule matematiche in linea (tra dollari, per esempio `$`⟨*numero*⟩`$`) o li si scrive nell'argomento del comando `\num` del pacchetto siunitx.

I numeri minuscoli presentano tratti ascendenti e discendenti per integrarsi al meglio con i caratteri minuscoli del font corrente, e sono indicati in tutti gli altri casi. Di solito, per ottenere i numeri minuscoli non c'è bisogno di fare nulla, perché spesso il font scelto li usa per impostazione predefinita. Si consiglia in ogni caso di fare una semplice prova per verificare la cosa: se i numeri predefiniti sono maiuscoli e si vogliono invece quelli minuscoli, la documentazione del font caricato dovrebbe indicare come farlo.

Si noti, infine, che anche se alcuni font (come i predefiniti Computer Modern) non distinguono tra numeri maiuscoli e minuscoli, si dovrebbe assumere da subito la buona abitudine di usare i dollari nei casi previsti: così facendo si potrà cambiare il font del documento senza dover modificare a mano ogni numero.

Spaziare le cifre

La corretta scrittura dei numeri *di cinque o più cifre* prevede uno spazio sottile fra ogni gruppo di tre cifre partendo da destra (come in 1 500 000). Per ottenerlo ci sono diverse possibilità: se i numeri da scrivere nel documento non sono molti, si può inserire a mano lo spazio sottile `\,` ; in caso contrario, risolve il problema il comando `\num` visto nel paragrafo precedente. Si osservi il risultato tipografico dei numeri negli esempi seguenti:

```
1500000 \\
$1500000$ \\
1\,500\,000 \\
$1\,500\,000$ \\
\num{1500000}
```

1500000
1500000
1 500 000
1 500 000
1 500 000

Come si può notare, se non altrimenti specificato `\num` stampa i numeri maiuscoli.

Si evitino il punto e la virgola per separare le cifre, perché questi due segni servono già da separatore decimale nel mondo anglosassone ed europeo rispettivamente. Lo spazio sottile è l'*unico* metodo universalmente corretto.

Numeri decimali

Nei numeri decimali il separatore fra parte intera e decimale è una virgola: «21,12». Si noti che in modo matematico LaTeX tratta la virgola come un normale segno di punteggiatura e aggiunge uno spazio extra dopo di essa, con un risultato insoddisfacente. siunitx caricato con l'opzione `output-decimal-marker={,}` e `\num` risolvono anche questo problema:

```
$21,12$ \\
\num{21,12}
```

21, 12
21,12

Si osservi che nel primo caso dopo la virgola si ha una spaziatura (leggermente) sbagliata: la scrittura corretta è la seconda.

A.4.3 Frazioni, percentuali, unità di misura

Le frazioni si esprimono in lettere, «tre quarti», a meno che non indichino una quantità numerica precisa. In questo caso si possono scrivere usando la barra, «3/4», o la forma frazionaria vera e propria, «$\frac{3}{4}$», ottenibile con il codice visto nel paragrafo 5.2.3 a pagina 62.

Nella scrittura del testo il simbolo di percento segue immediatamente il numero cui si riferisce senza spazi intermedi: «30%». Si ricorda che questo simbolo è un carattere speciale per i commenti nel codice sorgente: si ottiene in stampa premettendogli una barra rovescia: `\%`.

Le *quantità misurate* sono costituite da numeri seguiti da un'unità di misura espressa di solito con il simbolo relativo, come in «20 cm» o «15 kg». Si noti che il simbolo dell'unità di misura non vuole il punto e va separato dal numero cui si riferisce con uno spazio sottile. Per inserire correttamente le unità di misura del Sistema Internazionale si consiglia di usare il pacchetto siunitx o lo spazio sottile (si veda il paragrafo 5.12 a pagina 83).

A.4.4 Acronimi e abbreviazioni

Gli *acronimi* sono espressioni formate dalle lettere o sillabe iniziali delle parole di un'espressione che si vuole abbreviare, come GuIT, HTML o PDF. Dovrebbero essere composti interamente da maiuscole (si ammette anche il maiuscoletto), *senza* spaziature interne e punti d'abbreviazione. Se l'acronimo è entrato nell'uso e si può pronunciare come parola, è ammissibile (e spesso preferibile) scriverlo come tale: *Fiat* e *radar* (anziché FIAT e RADAR) vanno benissimo. Si consiglia di citare per esteso gli acronimi meno noti la prima volta che compaiono nel testo, mettendone la forma estesa fra parentesi: GuIT (Gruppo Utilizzatori Italiani di TeX e LaTeX).

Le *abbreviazioni* si ottengono invece dal troncamento di una parola mantenendone una o più lettere iniziali seguite dal punto. In casi particolari, come *sig.ra* o *prof.ssa*, l'abbreviazione comprende anche la parte finale della parola originale. Se l'abbreviazione dovesse cadere a fine enunciato (caso raro, forse possibile con il solo *ecc.*) il punto di abbreviazione funziona anche da punto fermo (cioè, ovviamente, non si scrivono due punti successivi).

B LINGUE ANTICHE

In quest'appendice si presentano gli strumenti essenziali per scrivere in latino e in greco, le due lingue fondamentali per chi si occupa di scienze umane. Se non indicato diversamente, si assume che la lingua principale del documento sia l'italiano.

B.1 SCRIVERE IN LATINO

Scrivere in latino con LaTeX non presenta particolari difficoltà: alfabeto e codifica di input sono gli stessi dell'italiano, e anche i font usati per il documento finito possono essere uguali.

Caricare babel indicando `latin` tra le lingue

```
\usepackage[latin,italian]{babel}
```

generalmente basta per scrivere *qualunque* testo latino (antico o contemporaneo, letterario e no) e per attivarne la corretta sillabazione. Per passare da una lingua all'altra si consiglia di personalizzare gli strumenti presentati nel paragrafo 3.2 a pagina 18 come spiegato nel paragrafo 9.2.3 a pagina 133 creandosi un comando `\latino`

```
\newcommand{\latino}[1]{\foreignlanguage{latin}{\em #1}}
```

e un ambiente `latino`

```
\newenvironment{latino}%
   {\begin{otherlanguage*}{latin}\em}%
   {\end{otherlanguage*}}
```

da usare nel modo consueto.

Di seguito si descrivono alcune *proprietates* di babel.

B.1.1 Marche prosodiche

L'attributo `withprosodicmarks` specificato subito dopo la chiamata di babel

```
\languageattribute{latin}{withprosodicmarks}
```

permette a LaTeX di riconoscere la marca della quantità sulle vocali *minuscole* (come spesso si trova nei testi didattici o di linguistica) che si ottiene semplicemente premettendo loro il carattere ^ per la quantità breve ˘ o il carattere = per quella lunga ¯:

```
\begin{otherlanguage*}{latin}
\ProsodicMarksOn\em
ros^a \\
ros=a
\ProsodicMarksOff
\end{otherlanguage*}
```

rosă
rosā

Si ricordi, però, di attivarne *effettivamente* la composizione racchiudendo il testo interessato con le *dichiarazioni* \ProsodicMarksOn e \ProsodicMarksOff, altrimenti il compositore restituisce esattamente quello che trova nel testo sorgente. Nei (rari) casi in cui sia una *maiuscola* a dover portare la quantità, si possono usare i comandi standard mostrati nella tabella 15 a pagina 49. In entrambi i casi LaTeX potrebbe sillabare in modo insoddisfacente o non sillabare per nulla le parole con la quantità segnata: si risolve il problema inserendo "| dove serve per indicare al programma i punti di cesura.

Si noti che non è permesso (né avrebbe senso) indicare la quantità di un dittongo legato in latino medievale (si veda il paragrafo seguente).

Per un difetto interno di babel, le dichiarazioni appena descritte non producono i loro effetti se non apportando due piccole modifiche al file babel.def (che contiene le definizioni del pacchetto), pena uno spazio indesiderato all'inizio del testo (come nell'esempio seguente). Di seguito si spiega come procedere:

- si copia sul proprio desktop il file babel.def dalla cartella in cui si trova (recuperabile dalle prime righe del *log* della compilazione, per esempio);

- si apre il file con l'editor in uso e si aggiunge un segno di percento *alla fine* delle righe 191 e 246 *senza lasciare ulteriori spazi* (si ricordi di impostare l'editor per visualizzare il numero delle righe, se non lo fosse già);

- si registrano le modifiche, si chiude il file modificato e lo si copia nella cartella $HOME/texmf/tex/generic/babel/ dell'albero personale, ricordandosi di eliminarlo quando gli autori di babel correggeranno ufficialmente il difetto in questione.

Chi non volesse apportare le modifiche appena descritte può usare in ogni caso i comandi standard ricordati più sopra in questo stesso paragrafo.

B.1.2 Latino medievale

Per dare al testo latino una patina medievaleggiante, l'autore *può* applicare tutte o solo alcune delle seguenti varianti:

- usare un unico segno, *u*, per *u* e *v* minuscole: *Nouembris* anziché *Novembris*, per esempio;

- usare un unico segno, *V*, per *U* e *V* maiuscole: *IVLIVS* anziché *Iulius*;

- usare le legature per i dittonghi *æ* e *œ* (minuscoli e maiuscoli), che nel latino classico erano *sempre* sciolti: *æ* (\ae), *œ* (\oe), *Æ* (\AE), *Œ* (\OE).

L'opzione latin di babel definisce un paio di automatismi utili per scrivere in questa varietà di latino:

- gestisce correttamente la conversione da minuscolo a maiuscolo nelle stringhe automatiche come i titoli di sezione: un titolo come \chapter {C\ae sar et Heluetii}, per esempio, nella corrispondente testatina maiuscola diventa *CÆSAR ET HELVETII*;

- ridefinisce \prefacename per ottenere *Præfatio* anziché *Praefatio* (le altre parole fisse non subiscono variazioni, si veda il paragrafo seguente).

Si seleziona questo tipo di scrittura caricando l'attributo medieval subito dopo la chiamata di babel:

```
\languageattribute{latin}{medieval}
```

Anche usando l'ortografia medievale si possono risolvere *eventuali* sillabazioni problematiche come già spiegato nel paragrafo precedente.

B.1.3 Documenti interamente in latino

Chi dovesse scrivere un documento completamente in latino può contare su qualche agevolazione in più. L'opzione `latin` di babel, infatti:

- converte automaticamente la data prodotta da `\today` nella corrispondente data "latina", nella forma XIX Februarii MMXII;
- traduce in latino tutte le parole fisse delle classi di documento standard: *Capitolo* diventa *Caput*, per esempio.

Si fornisce infine il preambolo tipo per comporre un documento in italiano e latino o nel solo latino (le righe sono numerate per maggiore chiarezza). Si noti che:

- va indicata obbligatoriamente *una sola* tra le righe 5 e 6 a seconda della o delle lingue del documento;
- le righe 8 e 9, invece, servono soltanto per sfruttare le rispettive specialità di `latin` e si possono anche caricare insieme.

```
1  \documentclass[a4paper]{article}
2  \usepackage[T1]{fontenc}
3  \usepackage[utf8]{inputenc}
4
5  \usepackage[latin,italian]{babel}
6  \usepackage[latin]{babel}
7
8  \languageattribute{latin}{withprosodicmarks}
9  \languageattribute{latin}{medieval}
10
11 \begin{document}
12
13 ...
14
15 \end{document}
```

B.2 SCRIVERE IN GRECO

Scrivere in greco con LaTeX richiede qualche attenzione: non soltanto è diverso l'alfabeto, ma lo sono anche la codifica di output e i font che il sistema usa per comporre il documento finito.

Prima di mettersi a scrivere, l'autore deve scegliere almeno due cose:

1. il *sistema ortografico* che intende seguire;
2. il *metodo di scrittura* del testo greco nel file sorgente.

LaTeX gestisce correttamente due sistemi ortografici per scrivere in questa lingua:

- il sistema *politonico* ("con molte intonazioni"), caratterizzato dalla presenza di tre accenti e numerosi altri segni diacritici, con cui si può scrivere il greco antico;

- il sistema *monotonico* ("con una sola intonazione"), caratterizzato dalla presenza di un solo accento, con cui si può scrivere il greco moderno.

Nel testo sorgente, infine, si possono scrivere le parti in greco in due modi:

- *traslitterando* il testo greco secondo una particolare corrispondenza tra i caratteri;
- scrivendo *direttamente* i caratteri greci.

Le possibilità appena descritte si possono combinare per un totale di quattro. Se non diversamente indicato, gli esempi di questo paragrafo sono in greco antico politonico.

L'opzione `greek` tra le lingue di babel

```
\usepackage[greek,italian]{babel}
```

permette di scrivere solo in greco monotonico. In questo modo si caricano automaticamente i font greci predefiniti CB (di Claudio Beccari, compresi nelle moderne distribuzioni di LaTeX, ma ne sono disponibili molti altri, alcuni dei quali hanno anche i caratteri latini) e si attivano la codifica di output `LGR` (*Local GReek*) e la corretta sillabazione delle parole. Si noti che per la traduzione delle parole fisse in greco antico in un documento interamente in greco (si veda il paragrafo B.2.3 a pagina 164) e la corretta interpretazione del carattere ~ (si veda il paragrafo B.2.1 a fronte) bisogna specificare *anche* l'attributo `polutoniko` subito dopo la chiamata di babel:

```
\languageattribute{greek}{polutoniko}
```

Si passa da una lingua all'altra con gli strumenti già esaminati nel paragrafo 3.2 a pagina 18 come, per esempio:

```
\dots testo in italiano.

\begin{otherlanguage*}{greek}
Th| p'anta dido'ush| ka`i
>apolambano'ush| f'usei <o
pepaideum'enos ka`i a>id'hmwn
l'egei; <<d`os, <`o j'eleis,
>ap'olabe, <`o j'eleic>>.
L'egei d`e to~uto o>u
katajrasun'omenos, >all`a
peijarq~wn m'onon ka`i
e>uno~wn a>ut~h|.
\end{otherlanguage*}

Altro testo in italiano\dots
```

…testo in italiano.
Τῃ πάντα διδούσῃ καὶ ἀπολαμβανούσῃ φύσει ὁ πεπαιδευμένος καὶ αἰδήμων λέγει· "δὸς, ὃ θέλεις, ἀπόλαβε, ὃ θέλεις". Λέγει δὲ τοῦτο οὐ καταθρασυνόμενος, ἀλλὰ πειθαρχῶν μόνον καὶ εὐνοῶν αὐτῇ.
Altro testo in italiano…

e si può inserire un testo breve o brevissimo in un capoverso scrivendolo nell'argomento di un comando `\greco` (leggermente diverso dall'analogo `\latino`) definito ad hoc:

```
\newcommand{\greco}{\foreignlanguage{greek}}
```

da usare come segue:

```
In lettere minuscole, \TeX{}
si scriverebbe \greco{teq}.
È la radice della parola greca
\greco{t'eqnh} che vuol dire
''arte'' e ''mestiere''.
```

In lettere minuscole, TeX si scriverebbe τεχ. È la radice della parola greca τέχνη che vuol dire "arte" e "mestiere".

| \ ̓ ! 1 " ¨ 2 £ 3 $ 4 % 5 & 6 / 7 (8) 9 = 0 ? ; ' ´ ^ ì

Q X q χ W Ω w ω E Ε e ε R Ρ r ρ T Τ t τ Y Ψ y ψ U Υ u υ I Ι i ι O Ο o ο P Π p π é è [* +]

A Α a α S Σ s σ D Δ d δ F Φ f φ G Γ g γ H Η h η J Θ j ϑ K Κ k κ L Λ l λ ç ò @ ° à # § ù

> ̓ < ̔ Z Ζ z ζ X Ξ x ξ C c ς V v B Β b β N Ν n ν M Μ m μ ; · , : . _ -

Figura 15: Corrispondenza tra la tastiera italiana e i caratteri greci.

B.2.1 Traslitterare i caratteri

Greco antico

Come si può osservare negli esempi precedenti, il testo sorgente "in greco" è un insieme di caratteri latini e altri segni (tutti conosciuti, ma qui con funzioni diverse) che poco assomiglia a ciò che si vede a destra. Lo si è scritto, infatti, con il metodo più immediato per chi possiede una tastiera italiana, cioè la *traslitterazione dei caratteri*. La figura 15 mostra (in blu) come l'opzione `greek` di babel interpreta i caratteri latini (alfabetici e non) che hanno una corrispondenza in greco. Tutti gli altri, compresi i caratteri speciali di LaTeX, sono uguali nelle due lingue (per la tilde, però, si veda sotto). Si noti che la lettera italiana *v* non ha un corrispettivo greco perché è una specie di "carattere greco speciale": la scrittura `sv`, infatti, è l'unico modo per ottenere un sigma iniziale o mediale isolato σ, altrimenti interpretato dal programma sempre come finale ς.

La tabella 47 nella pagina successiva raccoglie i segni di punteggiatura e i diacritici necessari per scrivere in greco. Si noti che:

- I font greci predefiniti permettono di combinare i diacritici in *tutti* i modi previsti dalla grammatica, premettendoli a una lettera *in una sequenza qualsiasi*.
- L'unico diacritico da posporre alla lettera è lo ι sottoscritto.
- Il carattere `"` nel testo sorgente restituisce un apostrofo soltanto se seguito da uno spazio (mantenuto anche nel testo composto, a differenza dell'italiano): `"␣`.
- Il carattere `~`, che in greco produce l'accento circonflesso, potrebbe *talvolta* produrre il consueto spazio indivisibile (che in greco si ottiene con `\nobreakspace`), nelle seguenti sequenze: `~`, `~"`, `"~`, `~<`, `<~`, `~>` e `>~`. Si risolve il problema premettendo a ciascuna una barra rovescia e caricando nel preambolo il file `lgraccents-glyphs.def` con il comando `\input`.
- Il greco antico non prevedeva nella scrittura apici e virgolette, che si possono comunque ottenere con i codici `‘‘ ’’` per gli apici, `(( ))` per le virgolette caporali e `<< >>` per quelle inglesi.

L'opzione `greek` di babel definisce ulteriori caratteri utili per scopi particolari:

- i caratteri minuscoli per indicare alcuni numeri (il *qoppa* ϙ ottenuto con `\qoppa` per il numero 6, lo *stigma* ϛ con `\stigma` per il 90 e il *sampi* ͳ con `\sampi` per il 900); quelli per il *digamma* minuscolo ϝ, ottenuto

Tabella 47: Diacritici e punteggiatura del greco antico.

Segno		Codice	Esempio	Risultato
Accento	acuto	`'`	`'w`	ώ
	grave	`` ` ``	`` `w ``	ὼ
	circonflesso	`~`	`~w`	ῶ
Spirito	dolce	`>`	`>w`	ὠ
	aspro	`<`	`<w`	ὡ
Coronide		`>`	`t>~alla`	τἆλλα
Dieresi		`"`	`"i`	ϊ
ι sottoscritto		`|`	`w|`	ῳ
Apostrofo		`"␣`	`d" >eg'w`	δ’ ἐγώ
Punto	in alto	`;`		·
	e virgola	`?`		;

con `\ddigamma` e maiuscolo Ϝ, ottenuto con `\Digamma`; il carattere ϛ, ottenuto con `\vardigamma`;

- l’apice ʹ e il pedice ͵ per scrivere i numerali, ottenuti rispettivamente con `\anwtonos` e `\katwtonos`;
- i due comandi `\greeknumeral` e `\Greeknumeral`, che producono automaticamente il numero arabo scritto nel loro argomento (da 1 a 999 999, in greco lo zero non esisteva) nel corrispondente numero greco secondo il sistema di numerazione alfabetico, minuscolo e maiuscolo rispettivamente (ma si noti che qoppa, stigma e sampi vengono stampati *sempre* minuscoli):

```
\begin{otherlanguage*}{greek}
\greeknumeral{996} \\
\Greeknumeral{123}
\end{otherlanguage*}
```

ϡϟϛʹ
ΡΚΓʹ

Greco moderno

Dai primi anni Ottanta del secolo scorso, in Grecia il sistema politonico ha lasciato il posto a quello monotonico, che nella varietà scritta presenta un solo accento (dipende dalle scelte tipografiche degli editori, ma di solito è quello acuto).

Quest’ortografia, dunque, è molto più semplice dell’altra, bastando soltanto altri due caratteri oltre alle lettere e ai consueti moderni segni d’interpunzione:

- `'` per l’accento acuto;
- `"␣` (lo spazio, evidenziato, è necessario) per ottenere l’apostrofo, oppure `"` seguito da ι/Ι o υ/Υ per ottenere la dieresi sopra di essi.

Si dichiara nel preambolo caricando babel *senza alcun attributo*:

```
\usepackage[greek,italian]{babel}
```

e un testo in questa varietà di greco appare così:

```
\begin{otherlanguage*}{greek}
Pr'epei na skefto'ume mia
sun'arthsh ths opo'ias
gnwr'izoume 'oti up'arqei
to olokl'hrwma.
\end{otherlanguage*}
```

Πρέπει να σκεφτούμε μια συνάρτηση της οποίας γνωρίζουμε ότι υπάρχει το ολοκλήρωμα.

Si noti che nonostante greek preveda tre distinti schemi di sillabazione per il greco (due per il greco politonico, antico e moderno, e uno per il greco monotonico), di fatto babel usa *sempre* quello per il greco politonico antico.

B.2.2 Scrivere in greco nel sorgente

Il modo appena descritto per scrivere in greco non è particolarmente difficile: la corrispondenza di lettere e diacritici sulla tastiera si memorizza in poco tempo, ma la traslitterazione rende comunque il testo sorgente difficile da leggere, specie per chi il greco lo conosce.

Quando in un documento le parti in greco sono lunghe o lunghissime (o quando il documento è in greco per intero) poterle scrivere direttamente in lingua diventa una comodità quasi irrinunciabile. I metodi per farlo sono più d'uno: si può usare una tastiera bilingue o una tastiera virtuale che non richieda di ricordare tutti gli abbinamenti, oppure si può ridefinire allo scopo i tasti di una tastiera italiana (con i modi propri di ogni sistema operativo). Qualunque metodo si scelga tra quelli spiegati in questo paragrafo, l'editor in uso *deve* supportare pienamente la codifica Unicode.

Per comporre il documento *con LaTeX* usando i font greci predefiniti, a utf8 va sostituita la variante utf8x subito dopo aver caricato il pacchetto ucs. Scritto in greco, l'esempio iniziale diventa:

```
\documentclass[a4paper]{article}
\usepackage[T1]{fontenc}
\usepackage{ucs}
\usepackage[utf8x]{inputenc}
\usepackage[greek,italian]{babel}
\languageattribute{greek}{polutoniko}

\begin{document}

\begin{otherlanguage*}{greek}
Τῃ πάντα διδούσῃ καὶ ἀπολαμβανούσῃ φύσει ὁ πεπαιδευμένος καὶ αἰδήμων
λέγει· “δὸς, ὃ θέλεις, ἀπόλαβε, ὃ θέλεις”. Λέγει δὲ τοῦτο οὐ
καταθρασυνόμενος, ἀλλὰ πειθαρχῶν μόνον καὶ εὐνοῶν αὐτῃ.
\end{otherlanguage*}

\end{document}
```

L'alternativa è il programma XƎLaTeX (di solito si può scegliere tra i motori gestiti dall'editor in uso), che richiede un preambolo come il seguente:

```
\documentclass[a4paper]{article}
\usepackage{fontspec}
\setmainfont{Linux Libertine O}
\usepackage{polyglossia}
\setmainlanguage{italian}
\setotherlanguage[variant=ancient]{greek}

\begin{document}

\begin{greek}
Τῃ πάντα διδούσῃ καὶ ἀπολαμβανούσῃ φύσει ὁ πεπαιδευμένος καὶ αἰδήμων
λέγει· “δὸς, ὃ θέλεις, ἀπόλαβε, ὃ θέλεις”. Λέγει δὲ τοῦτο οὐ
καταθρασυνόμενος, ἀλλὰ πειθαρχῶν μόνον καὶ εὐνοῶν αὐτῃ.
\end{greek}

\end{document}
```

Si noti che:

- il pacchetto fontenc è sostituito da fontspec, che gestisce i font del sistema in modo molto potente;
- il pacchetto inputenc *non* va caricato, perché XƎLaTeX lavora con la codifica UTF-8 per impostazione predefinita;
- \setmainfont imposta il font principale del documento con le eventuali opzioni (qui si è scelto il font Linux Libertine, già presente nel sistema e dotato di una collezione completa di font greci);
- il pacchetto babel è sostituito da polyglossia;
- \setmainlanguage imposta la lingua principale del documento con le eventuali opzioni;
- \setotherlanguage carica l'altra lingua del documento con le eventuali opzioni, in questo caso il greco antico nella variante ancient, cioè politonico (si possono caricare in una volta sola più lingue *senza opzioni* con il comando \setotherlanguages);
- l'ambiente greek equivale a otherlanguage*{greek}.

A differenza di babel, polyglossia carica *effettivamente* la sillabazione corrispondente alla variante indicata: ancient (o poly) per il greco politonico, mono per quello monotonico.

Entrambe le possibilità descritte in questo paragrafo, però, soffrono di una limitazione:

- i pacchetti ucs e biblatex non sono attualmente compatibili tra loro;
- usando XƎLaTeX, nelle parti in italiano microtype permette solo la protrusione dei caratteri, ma non la microgiustificazione e altre finezze.

B.2.3 Documenti interamente in greco

Chi dovesse scrivere un documento completamente in greco (seguendo indifferentemente l'una o l'altra ortografia) può contare su qualche agevolazione in più. L'opzione greek di babel, infatti:

- definisce due comandi per convertire automaticamente la data corrente nella corrispondente data greca: \dategreek la stampa con i numerali arabi (19 Φεβρουαρίου 2012), \Grtoday con i numerali alfabetici (ΙΘʹ Φεβρουαρίου ͵ΒΙΒʹ);
- traduce in greco le parole fisse delle classi di documento standard in accordo con l'ortografia dichiarata nel preambolo: *Indice* diventa Εὑρετήριο in greco politonico, ma Ευρετήριο in greco monotonico, per esempio;
- definisce altri due comandi utili: \euro, che produce il simbolo dell'euro € e \permill, che produce il simbolo di per mille ‰.

Si fornisce infine il preambolo tipo per comporre un documento in italiano e greco o nel solo greco *con LaTeX* (le righe sono numerate per maggiore chiarezza). Si noti che:

- va indicata obbligatoriamente *una sola* tra le righe 5 e 6 a seconda della o delle lingue del documento;

- la riga 8, invece, va messa subito dopo babel soltanto per avere la corretta traduzione delle parole fisse e la corretta interpretazione di ~.

```
1  \documentclass[a4paper]{article}
2  \usepackage[T1]{fontenc}
3  \usepackage[utf8]{inputenc}
4
5  \usepackage[greek,italian]{babel}
6  \usepackage[greek]{babel}
7
8  \languageattribute{greek}{polutoniko}
9
10 \begin{document}
11
12 ...
13
14 \end{document}
```

Naturalmente, anche per XeLaTeX valgono le osservazioni di questo paragrafo: basterà modificare il preambolo esemplificato nel paragrafo B.2.2 a pagina 163 impostando il greco come lingua principale del documento con `\setmainlanguage[⟨opzioni⟩]{greek}`.

B.2.4 Specialità: teubner

Chi avesse la necessità di comporre documenti in greco di altro tipo (in particolare linguistici, filologici e letterari), può usare il pacchetto teubner, che comprende una collezione di caratteri inclinati particolarmente raffinati, disegnati originariamente nella tipografia Teubner di Lipsia (in onore della quale è stato scelto il nome del pacchetto), e tra le altre cose permette di regolare molto finemente le combinazioni dei diacritici sulle lettere e le legature tra i caratteri.

Si raccomanda infine di caricare teubner *dopo* amsmath, se dovesse servire anche quest'ultimo pacchetto. Per maggiori dettagli si rimanda alla sua documentazione.

C VIDEOPRESENTAZIONI

Quest'appendice, pensata come naturale complemento visivo alla redazione della tesi di laurea, fornisce gli strumenti fondamentali per realizzare una videopresentazione con LaTeX. Si rimanda chi voglia approfondire l'argomento alla corposa e dettagliata documentazione del pacchetto e a [Pantieri e Gordini, 2009], da cui queste note sono tratte.

C.1 INTRODUZIONE

Nelle prossime pagine si descriverà nei suoi elementi essenziali beamer, una classe di documento molto potente e altamente versatile dedicata alle videopresentazioni. Fra i suoi punti di forza ci sono la gestione semplice degli effetti dinamici tra le diapositive, la possibilità di sezionare il testo con i comandi standard, la piena compatibilità con gli altri pacchetti e un ventaglio di personalizzazioni virtualmente illimitato.

Si noti che beamer carica automaticamente i pacchetti hyperref, graphicx, amsmath, amssymb, amsthm.

Qualche consiglio

Una presentazione efficace è il risultato dell'esposizione chiara di contenuti essenziali, possibilmente riprodotti in diapositive poco elaborate: in questo beamer aiuta molto, ma da solo non basta. I consigli seguenti dovrebbero aiutare l'autore:

- scrivere e parlare il più semplicemente possibile;
- ridurre il contenuto della diapositiva al minimo indispensabile (il pubblico deve impiegare il proprio tempo ascoltando chi parla e non leggendo);
- proiettare una diapositiva al minuto, non di più;
- suddividere la presentazione in frazioni di circa 5-7 minuti l'una;
- usare un font senza grazie e di corpo molto grande (si deve poter leggere la diapositiva senza fatica anche nelle ultime file);
- evitare capoversi giustificati (difficili da ottenere con caratteri grandi e non necessari, se il testo è *davvero* essenziale);
- usare lo stesso formato per tutte le diapositive.

C.2 FONDAMENTALI

In linea generale, la struttura di una presentazione realizzata con beamer consiste in una serie di *quadri* (*frame*, in inglese), ciascuno dei quali è composto da una o più *diapositive* (*slide*).

C.2.1 Per cominciare

L'esempio

```
\documentclass{beamer}

\title{La nostra prima presentazione}
\author{Lorenzo Pantieri \and Tommaso Gordini}
\date{1 maggio 2011}

\begin{document}

% quadro 1
\begin{frame}
\maketitle
\end{frame}

% quadro 2
\begin{frame}
\frametitle{Un esempio}
\begin{itemize}
\item<1-> Mane
\item<2-> Tekel
\item<3-> Fares
\end{itemize}
\end{frame}

\end{document}
```

riproduce il codice della semplice presentazione di quattro diapositive in due quadri mostrata nella figura 16 nella pagina successiva: il primo quadro ne contiene solo il titolo generale; il secondo un elenco di tre elementi. Si osservi che le cose ancora sconosciute sono poche:

- l'ambiente `frame` racchiude un quadro;
- il comando `\frametitle` produce il titolo del quadro: *ogni* quadro dovrebbe averne uno;
- se le diapositive di un quadro sono più d'una, si mettono in un ambiente `itemize`, dove ogni `\item` produce una diapositiva;
- le espressioni racchiuse tra i segni di < e > verranno spiegate nel paragrafo C.2.3 a pagina 173.

Anche in questo caso, infine, un uso oculato dei commenti permette di individuare facilmente le varie parti della presentazione nel testo sorgente.

C.2.2 Una presentazione articolata

Il professor Euclide dell'Università di Alessandria d'Egitto è pronto per rivelare al mondo la sua ultima scoperta: i numeri primi sono infiniti. Euclide parlerà durante il VII Simposio Internazionale sui Numeri Primi dove avrà a disposizione venti minuti, cinque dei quali riservati alle domande del pubblico. La presentazione che ha preparato è la seguente:

```
\documentclass{beamer}
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}
\usepackage[italian]{babel}
```

(a) Quadro I: diapositiva 1/1.
(b) Quadro II: diapositiva 1/3.
(c) Quadro II: diapositiva 2/3.
(d) Quadro II: diapositiva 3/3.

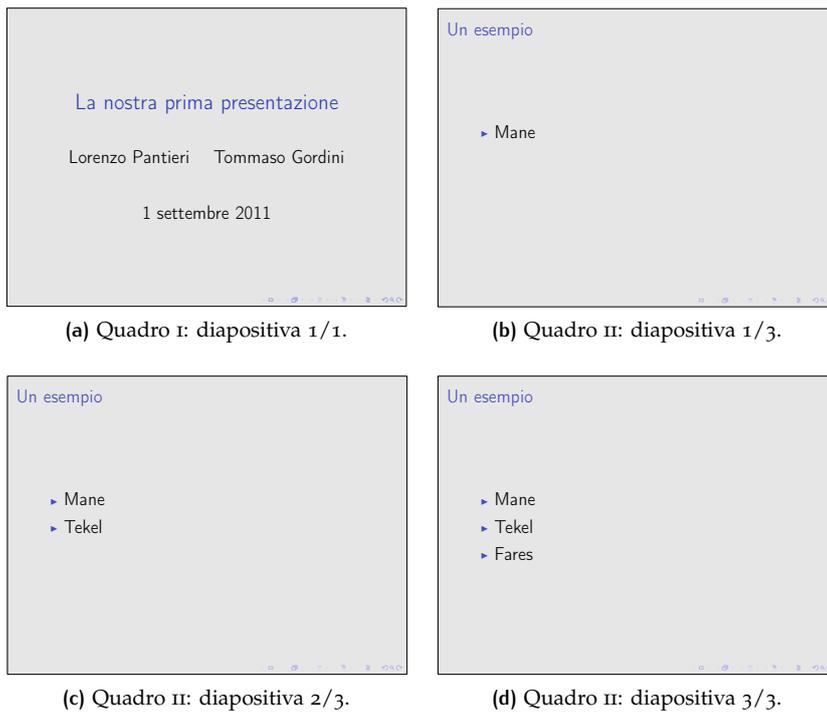



Figura 16: Presentazione semplice.

```
\title{I numeri primi sono infiniti}
\author[Euclide]{Euclide di Alessandria \\
\texttt{euclide@alessandria.edu}}
\date[VII SINP]{VII Simposio Internazionale sui Numeri Primi}
\institute[UniAlessandria]{Università di Alessandria}
\logo{\includegraphics[width=0.2\textwidth]{sigillo}}

\usetheme{AnnArbor}
\useoutertheme[right]{sidebar}
\setbeamercovered{dynamic}

\theoremstyle{definition}
\newtheorem{definizione}{Definizione}
\theoremstyle{plain}
\newtheorem{teorema}{Teorema}

\begin{document}

% quadro 1
\begin{frame}
\maketitle
\end{frame}

% quadro 2
\begin{frame}
\frametitle{Piano della presentazione}
\tableofcontents
\end{frame}

% quadro 3
\section{Introduzione}
\begin{frame}
```

```
\frametitle{Che cosa sono i numeri primi?}
\begin{definizione}
Un \alert{numero primo} è un intero $>1$ che ha esattamente
due divisori positivi.
\end{definizione}
\end{frame}

% quadro 4
\section{L'infinità dei primi}
\begin{frame}
\frametitle{I numeri primi sono infiniti}
\framesubtitle{Ne diamo una dimostrazione diretta}
\begin{teorema}
Non esiste un primo maggiore di tutti gli altri.
\end{teorema}
\pause
\begin{proof}
\begin{enumerate}[<+->]
\item Sia dato un elenco di primi.
\item Sia $q$ il loro prodotto.
\item Allora $q+1$ è divisibile per un primo $p$
      che non compare nell'elenco. \qedhere
\end{enumerate}
\end{proof}
\end{frame}

% quadro 5
\section{Problemi aperti}
\begin{frame}
\frametitle{Che cosa c'è ancora da fare?}
\begin{block}{Problemi risolti}
Quanti sono i numeri primi?
\end{block}
\begin{block}{Problemi aperti}
Un numero pari $>2$ è sempre la somma di due primi?
\end{block}
\end{frame}

\end{document}
```

è una presentazione di otto diapositive in cinque quadri (mostrata nella figura 17 nella pagina successiva) che di seguito si analizza negli elementi di novità.

Preambolo

Contiene come di consueto le istruzioni generali del documento. Si noti che:

- gli argomenti facoltativi di `\author` e `\date` mettono il loro contenuto in punti particolari della diapositiva (di solito in basso).
- `\institute` (il cui argomento facoltativo si comporta come descritto sopra) inserisce nella diapositiva il nome dell'istituzione di afferenza e `\logo` ne mette il logo. (Se lo si desidera, si può anche inserire una figura nella diapositiva inziale scrivendo nel preambolo l'istruzione `\titlegraphic{\includegraphics{`⟨*figura*⟩`}}`.)
- Con `\usetheme` si sceglie il *tema* della presentazione, cioè l'aspetto generale delle diapositive per colori e disposizione degli elementi, e con

(a) Quadro I: diapositiva 1/1.

(b) Quadro II: diapositiva 1/1.

(c) Quadro III: diapositiva 1/1.

(d) Quadro IV: diapositiva 1/4.

(e) Quadro IV: diapositiva 2/4.

(f) Quadro IV: diapositiva 3/4.

(g) Quadro IV: diapositiva 4/4.

(h) Quadro V: diapositiva 1/1.

Figura 17: Presentazione complessa.

`\useoutertheme` si scelgono ulteriori elementi di contorno (si veda il paragrafo C.3 a pagina 177).

- `\setbeamercovered` regola l'aspetto del testo nella proiezione. Accetta tre valori: con `transparent` il testo non ancora proiettato è (semi)trasparente (quel tanto che basta a suggerire a chi parla come proseguire l'esposizione); con `invisible` (predefinito) è completamente nascosto; con `dynamic` è tanto più trasparente quanto più tempo deve rimanere nascosto, e viceversa.

Primo quadro

Composto di una sola diapositiva (figura 17a), contiene il titolo della presentazione. In basso a destra si vede il contatore dei quadri in forma di frazione (1/5): quello proiettato, dunque, è il primo di cinque quadri in totale.

Secondo quadro

Composto di una sola diapositiva (figura 17b), contiene l'indice generale dell'esposizione, utile all'inizio per illustrare il piano della proiezione. L'indice è riportato esattamente sulla *barra di navigazione* laterale, le cui voci (cliccabili per portarsi rapidamente da un punto all'altro della presentazione) si illuminano progressivamente all'esaurirsi dei quadri.

Intermezzo

Tra il secondo e il terzo quadro c'è un comando di sezionamento standard che va dato, si noti bene, *prima* che il quadro cominci. In beamer questi comandi servono solo a comporre l'indice generale: il contenuto del loro argomento non verrà *mai* mostrato altrove. Si ricordi che una presentazione non è un documento tradizionale, per cui si raccomanda di valutarne attentamente la struttura e di non sezionarla eccessivamente.

Terzo quadro

Composto di una sola diapositiva (figura 17c), contiene una definizione in cui due parole sono evidenziate con il comando `\alert`, che marca il proprio argomento diversamente dai comandi standard (in blu, in questo caso).

Quarto quadro

È il quadro più articolato, composto di quattro diapositive (figure 17d-g). Si noti che qualunque sia il numero di diapositive che lo compongono, il suo contenuto va scritto *una sola volta*: le tecniche di esposizione incrementale spiegate nel prossimo paragrafo si occuperanno di dilazionarne la proiezione.

Quinto quadro

Composto di una sola diapositiva, evidenzia il proprio contenuto in due *blocchi* (i due ambienti `block`), per i quali si rimanda al paragrafo C.2.4 a pagina 174.

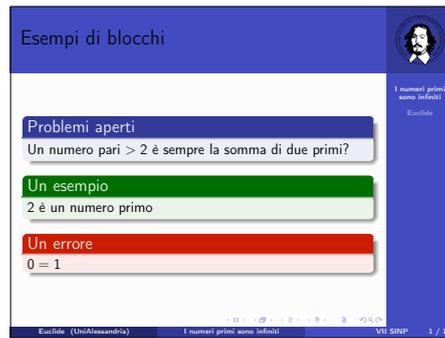



Figura 18: Esempi di blocchi.

C.2.3 Esposizione incrementale

Rivelare progressivamente il contenuto di un quadro (specialmente un ragionamento svolto per punti, ma anche un elenco) proiettandone gli elementi uno alla volta può aiutare il pubblico a seguire meglio il discorso e a limitare il carico d'attenzione. Con `beamer` questo effetto si ottiene in diversi modi, anche se si raccomanda di non abusarne e di contenerlo ai casi davvero necessari.

Il metodo più semplice è frazionare il contenuto del quadro dando nei punti opportuni altrettanti comandi `\pause`. Ogni diapositiva mostrerà il testo corrente più tutto quello precedente, e così via.

Il secondo metodo è dare dopo ogni `\item` il comando `\onslide`, la cui sintassi generale è

```
\onslide<⟨intervallo⟩>{⟨testo⟩}
```

dove il ⟨*testo*⟩ verrà mostrato solo nelle diapositive indicate nell'⟨*intervallo*⟩. Nelle altre, anche se non presente sullo schermo, avrà comunque lo spazio corrispondente. Per esempio, un elenco i cui punti scompaiano uno dopo l'altro dopo essere stati proiettati si ottiene con:

```
\begin{frame}
\begin{itemize}
\item \onslide<1>{Mane}
\item \onslide<2>{Tekel}
\item \onslide<3>{Fares}
\end{itemize}
\end{frame}
```

Il terzo metodo consiste nello scrivere subito dopo ciascun `\item` un'espressione che nella sua forma più completa è:

```
<⟨numero⟩>-<⟨numero⟩>
```

dove:

- `<`⟨*numero*⟩`>` è il numero progressivo della diapositiva nella presentazione;
- un semplice numero (`<4>`, per esempio) fa proiettare l'elemento una sola volta nella quarta diapositiva del quadro;
- un intervallo numerico separato con un trattino (`<4-6>`, per esempio) mantiene l'elemento visibile nelle diapositive dalla quarta alla sesta comprese, ma non prima né dopo;
- non scrivere nulla prima del trattino (`<-4>`) equivale a scrivere `<1-4>`;

- non scrivere nulla dopo il trattino (`<4->`) rende visibile l'elemento dalla diapositiva in cui compare la prima volta (la quarta, in questo caso) fino all'ultima diapositiva del quadro (come nella prima presentazione analizzata).

Si poteva scrivere più sinteticamente il secondo quadro della prima presentazione con:

```
\begin{itemize}[<+->]
\item Mane
\item Tekel
\item Fares
\end{itemize}
```

C.2.4 Blocchi

In beamer un *blocco* è una cornice più o meno marcata che evidenzia la porzione di testo al proprio interno. La classe definisce tre ambienti standard per i blocchi, che vengono resi in modo diverso a seconda del tema caricato: `block` per i blocchi generici, `exampleblock` per gli esempi e `alertblock` per gli avvisi. Per esempio, in una presentazione con il tema `Madrid` il codice

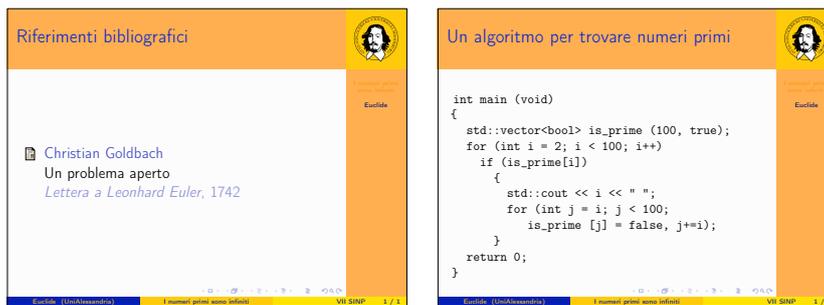
```
\begin{frame}
\begin{block}{Problemi aperti}
Un numero pari $>2$ è sempre la somma di due primi?
\end{block}

\begin{exampleblock}{Un esempio}
$2$ è un numero primo
\end{exampleblock}

\begin{alertblock}{Un errore}
$0=1$
\end{alertblock}
\end{frame}
```

produce la diapositiva mostrata nella figura 18 nella pagina precedente, in cui si può osservare che la differenza fra i blocchi è soltanto cromatica. Per le innumerevoli personalizzazioni possibili si rimanda alla documentazione della classe.

C.2.5 Bibliografia

Non dovrebbe *mai* essere necessario proiettare i riferimenti bibliografici del proprio lavoro. Nei rari casi in cui servisse, si può usare un codice come

(a) Diapositiva con riferimento bibliografico.

Un algoritmo per trovare numeri primi

```
int main (void)
{
  std::vector<bool> is_prime (100, true);
  for (int i = 2; i < 100; i++)
    if (is_prime[i])
      {
      std::cout << i << " ";
      for (int j = i; j < 100;
         is_prime [j] = false, j+=i);
      }
  return 0;
}
```

(b) Diapositiva con codice.

Figura 19: Diapositive particolari.

```
\begin{frame}
\frametitle{\refname}
\begin{thebibliography}{9}
\bibitem{goldbach:congettura} Christian Goldbach
\newblock Un problema aperto
\newblock \emph{Lettera a Leonhard Euler}, 1742
\end{thebibliography}
\end{frame}
```

che produce la diapositiva mostrata nella figura 19a, dove il comando standard `\newblock` separa i diversi elementi della bibliografia. Va da sé che in una presentazione non vanno *mai* usati nemmeno i comandi come `\cite` o analoghi, perché durante la proiezione non si ha alcun modo di verificare a quale opera si riferiscono.

c.2.6 Codici

Per scrivere del codice in una diapositiva, come mostra la figura 19b, si può usare il tradizionale ambiente `verbatim`, ricordandosi di dare a `frame` l'opzione `fragile`:

```
\begin{frame}[fragile]
\frametitle{Un algoritmo per trovare numeri primi}
\begin{verbatim}
  int main (void)
  {
    std::vector<bool> is_prime (100, true);
    for (int i = 2; i < 100; i++)
      if (is_prime[i])
        {
        std::cout << i << " ";
        for (int j = i; j < 100; is_prime [j] = false, j+=i);
        }
    return 0;
  }
\end{verbatim}
\end{frame}
```

(a) Due colonne centrate verticalmente.

(b) Due colonne allineate.

(c) Una figura e un elenco.

(d) Un elenco e una figura.

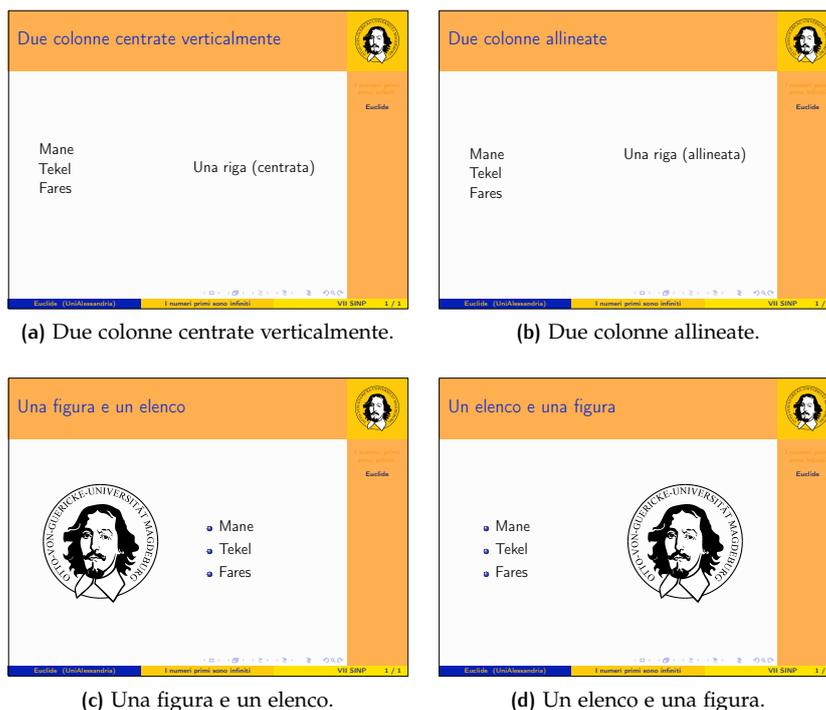



Figura 20: Diapositive strutturate su più colonne.

c.2.7 Disporre il contenuto su più colonne

Organizzare il contenuto di una diapositiva su più colonne torna utile soprattutto quando si devono inserire tabelle o figure con la propria descrizione, che in una presentazione va messa sempre *accanto* a esse. Gli ambienti mobili `table` e `figure`, infatti, non hanno ragione di essere usati: chi parla non dirà mai «come abbiamo visto nella figura n», perché *nessuno* durante la spiegazione si ricorderà quale figura corrisponde al numero n.

Le diverse colonne sono prodotte da altrettanti ambienti `column` (ai quali va *sempre* assegnata una larghezza) annidati in un unico ambiente `columns`. Il codice seguente

```
\begin{frame}
\begin{columns}
\begin{column}{0.4\textwidth}
Mane \\ Tekel \\ Fares
\end{column}

\begin{column}{0.4\textwidth}
Una riga (centrata)
\end{column}
\end{columns}
\end{frame}
```

produce la diapositiva mostrata nella figura 20a, nella quale le righe delle due colonne sono centrate verticalmente l'una rispetto all'altra (è l'impostazione predefinita). Se si vogliono le prime righe di ciascuna colonna allineate come nella figura 20b, invece, basta dare a `columns` l'opzione `[t]`.

Una figura (a sinistra) con accanto un elenco (a destra), come nella figura 20c si ottiene con

```
\begin{frame}
```

```
\begin{columns}
\begin{column}{0.4\textwidth}
\includegraphics[width=\columnwidth]{figura}
\end{column}

\begin{column}{0.4\textwidth}
\begin{itemize}
\item Mane
\item Tekel
\item Fares
\end{itemize}
\end{column}
\end{columns}
\end{frame}
```

L'effetto contrario (mostrato nella figura 20d a fronte) si ottiene invertendo il contenuto dei due ambienti `column`.

C.2.8 Stampare la presentazione

Assegnando a beamer l'opzione `handout` si ottiene facilmente anche una versione della presentazione adatta alla stampa (un *handout*) priva degli eventuali effetti dinamici inseriti per la proiezione. A questo scopo il pacchetto pgfpages permette di raccogliere in una stessa pagina 2, 4, 8 o 16 diapositive caricandolo così:

```
\usepackage{pgfpages}
\pgfpagesuselayout{4 on 1}[a4paper,border shrink=5mm,landscape]
```

dove si chiede a LaTeX di mettere quattro diapositive in ogni pagina (`4 on 1`) di formato A4 disposta in orizzontale (`landscape`), con un piccolo spazio (`border shrink`) di 5 mm attorno a ciascuna diapositiva. (L'opzione `landscape` è richiesta soltanto se in una pagina si vogliono 4 o 16 diapositive, non negli altri casi.)

C.3 PRESENTAZIONI PERSONALIZZATE

In questa sezione si daranno soltanto alcuni esempi delle numerose possibilità che beamer offre per personalizzare le proprie presentazioni.

Il *tema* gestisce l'aspetto grafico della presentazione definendo i colori che appariranno nelle diapositive, il formato di elenchi e blocchi e infine l'aspetto e la disposizione degli elementi di contorno. Si noti che tutti questi elementi possono essere regolati uno per uno (la documentazione della classe spiega come farlo), moltiplicando le possibili personalizzazioni. Se non dovessero bastare i temi precaricati in beamer, se ne possono scaricare moltissimi altri da Internet o ancora crearsene di personali dal principio.

Si seleziona il tema scrivendo nel preambolo

```
\usetheme{⟨tema della presentazione⟩}
```

dove ⟨*tema della presentazione*⟩ è il nome del tema prescelto che, per cominciare, si può scegliere tra quelli predefiniti dalla classe identificati di regola dal nome di una città.

Non potendo qui riportarli tutti, si tenga presente che a grandi linee si possono dividere in:

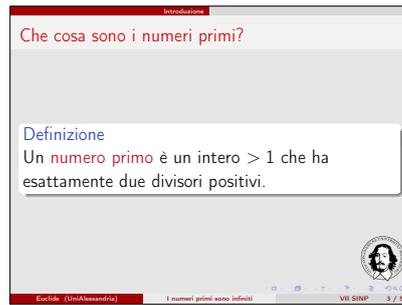
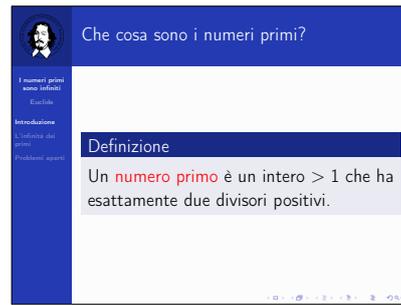
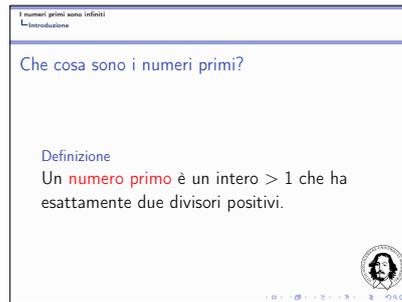
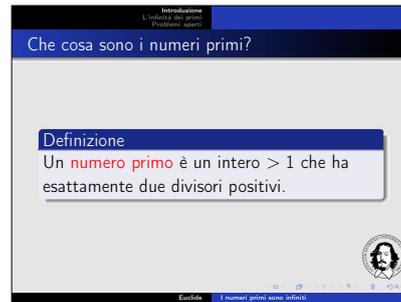
- temi senza barra di navigazione (`AnnArbor`, `CambridgeUS`, `Madrid`, per esempio);

(a) Il tema `CambridgeUS`.

(b) Il tema `Berkeley`.

(c) Il tema `Montpellier`.

(d) Il tema `Warsaw`.

Figura 21: Alcuni temi predefiniti di beamer.

- temi con barra di navigazione laterale (`Berkeley`, `Goettingen`, `Marburg`);
- temi con barra di navigazione ad albero (`Antibes`, `JuanLesPins`, `Montpellier`);
- temi con quadro di navigazione (`Berlin`, `Dresden`, `Warsaw`, `Singapore`).

Alcuni di essi vengono mostrati nelle diapositive della figura 21.

ACRONIMI

AMS American Mathematical Society

Fondata nel 1888, con svariate decine di migliaia di soci la "Società Matematica Americana" è una delle più importanti associazioni di matematici nel mondo. L'AMS ha sostenuto attivamente lo sviluppo di LaTeX ed è stata tra i primi organismi scientifici a sollecitare gli autori a scrivere con questo programma.

CTAN Comprehensive TeX Archive Network

La "rete di archivi completi di TeX" è, nel Web, il luogo di riferimento da cui scaricare software e materiale su TeX e LaTeX.

GuIT Gruppo Utilizzatori Italiani di TeX e LaTeX

È un'associazione senza fini di lucro con lo scopo di aumentare la diffusione di TeX e LaTeX in Italia attraverso la condivisione di informazioni legate al loro uso, conciliando il vantaggio dell'apprendimento con il piacere dell'insegnamento.

HTML Hyper Text Mark-up Language

Il "linguaggio di marcatura degli ipertesti" è un linguaggio di pubblico dominio inventato nel 1989 al CERN di Ginevra da Tim Berners-Lee per descrivere i documenti ipertestuali che popolano il Web.

ISO International Standard Organization

L' "organizzazione internazionale per le standardizzazioni" è la più importante associazione a livello mondiale per la definizione di standard tecnico-scientifici. Suoi membri sono gli organismi nazionali di standardizzazione di 157 Paesi del mondo.

JPEG Joint Photographic Experts Group

È il più usato standard di compressione delle immagini fotografiche. Esegue una compressione con perdita di informazioni (cioè di tipo *lossy*). Insieme con il PNG, è il formato standard delle immagini bitmap da inserire in un documento da comporre con LaTeX.

PDF Portable Document Format

È il formato di file più versatile per la stampa e la distribuzione elettronica, introdotto dalla Adobe Systems nel 1993 per rappresentare documenti indipendentemente dall'hardware e dal software usati per generarli o visualizzarli. Il PDF eredita molte delle funzioni del PostScript, un linguaggio di descrizione della pagina sviluppato dalla stessa azienda. È il formato standard delle immagini vettoriali da inserire in un documento da comporre con LaTeX.

PNG Portable Network Graphics

Creato nel 1995, è un formato di file particolarmente adatto per rappresentare disegni e icone. Esegue una compressione senza perdita di informazioni (cioè di tipo *lossless*). Con il JPG, è il formato standard delle immagini bitmap da inserire in un documento da comporre con LaTeX.

TUG TeX User Group

L'espressione ("gruppo di utenti di TeX") indica le associazioni di persone accomunate dalla passione per TeX e LaTeX sparse nei principali Paesi del mondo. Il loro scopo è quello di diffondere l'uso di questi due programmi e di fornire supporto alle rispettive comunità di utenti.

UNI Ente Nazionale Italiano di Unificazione

È un'associazione privata senza scopo di lucro che svolge attività normativa in tutti i principali settori tecnico-scientifici. Rappresenta l'Italia all'attività normativa dell'ISO.

URL Uniform Resource Locator

È una stringa di caratteri che identifica in modo univoco l'indirizzo di una qualunque risorsa in Rete. Ogni URL (per esempio `http://www.tug.org/texlive/`) si compone normalmente di tre parti: il protocollo usato per indirizzare la risorsa (`http`, nell'esempio considerato), il nome dell'*host* o del server o del dominio (`www.tug.org`), e infine il nome del file che contiene la risorsa (`/texlive/`).

UTF Unicode Transformation Format

Unicode è un sistema di codifica che assegna una combinazione di bit a ogni carattere indipendentemente da programma, piattaforma e lingua usati. La codifica UTF-8 (*Unicode Transformation Format 8 bit*) è una particolare realizzazione di Unicode.

WYSIWYG What You See Is What You Get

L'espressione indicata dall'acronimo ("ciò che vedi è ciò che ottieni") ha sostanzialmente due significati.

Il primo si riferisce al problema di ottenere in stampa testo e immagini che abbiano una disposizione grafica uguale a quella visualizzata sullo schermo del calcolatore. I primi software e le prime stampanti per uso domestico non davano risultati pienamente soddisfacenti e si superò il problema introducendo nuovi dispositivi e software (pionieri furono il sistema di codifica dei caratteri TrueType sviluppato dalla Apple e il programma Adobe TypeManager).

Con il tempo, il significato dell'acronimo si è esteso per analogia anche ad alcune problematiche connesse alla creazione dei documenti. Nei comuni elaboratori di testo (come per esempio Microsoft Word), l'autore agisce direttamente sul testo già composto così come appare sullo schermo, e ogni sua azione si traduce in un'immediata variazione di quel testo. Programmi di questo tipo vengono detti WYSIWYG (in questo secondo significato), e il tipo di composizione che adottano viene denominato "composizione sincrona". L'acronimo che si riferisce al concetto opposto è WYSIWYM.

WYSIWYM What You See Is What You Mean

L'acronimo ("ciò che vedi è ciò che intendi") è stato coniato espressamente per LaTeX in contrapposizione a WYSIWYG (nel senso di "programma di videoscrittura caratterizzato da una composizione sincrona").

La caratteristica che più differenzia LaTeX dagli altri elaboratori di testo è il fatto che per realizzare un documento con questo programma bisogna agire *in tempi diversi* per introdurre il testo e per comporlo. È la "composizione asincrona": prima si scrive il testo badando soltanto a contenuto e scansione logica; poi lo si dà "in pasto" a LaTeX, che lo compone e lo impagina per produrre il documento finito.

ELENCO DEI SITI INTERNET

CHARCO
Sito ufficiale di Charco.
http://www.marblesoftware.com/Marble_Software/Charco.html

CTAN
Sito ufficiale di CTAN.
http://www.ctan.org/

EMACS
Pagina Web di Emacs.
http://www.gnu.org/software/emacs/

ESCHER
Sito ufficiale di Maurits C. Escher.
http://www.mcescher.com/

GUIT
Sito ufficiale del GuIT.
http://www.guitex.org/

KILE
Pagina Web di Kile.
http://kile.sourceforge.net/

KNUTH
Pagina Web di Donald E. Knuth.
http://www-cs-faculty.stanford.edu/~knuth/

LAMPORT
Pagina Web di Leslie Lamport.
http://research.microsoft.com/en-us/um/people/lamport/

MACTEX
Pagina Web di MacTEX.
http://www.tug.org/mactex/

MIKTEX
Sito ufficiale di MiKTEX.
http://miktex.org/

SAROVAR
Catalogo LATEX online.
http://texcatalogue.sarovar.org/

TEXLIVE
Pagina Web di TEX Live.
http://www.tug.org/texlive/

TEXMAKER
Pagina Web di Texmaker.
http://www.xm1math.net/texmaker/

TEXSTUDIO
Pagina Web di TeXstudio.
http://sourceforge.net/projects/texstudio/

TEXSHOP
Pagina Web di TeXShop.
http://www.uoregon.edu/~koch/texshop/

TEXWORKS
Pagina Web di TEXworks.
http://www.tug.org/texworks/

BIBLIOGRAFIA

Beccari, Claudio

2011 *Introduzione all'arte della composizione tipografica con LaTeX*, http://www.guit.sssup.it/downloads/GuidaGuIT.pdf. (Citato alle pp. 3, 15, 41, 61.)

Bringhurst, Robert

1992 *The Elements of Typographic Style*, Hartley & Marks, Vancouver, (edizione italiana a cura di Lucio Passerini, *Gli Elementi dello Stile Tipografico*, Sylvestre Bonnard, Milano 2001). (Citato a p. 30.)

Cevolani, Gustavo

2006 «Norme tipografiche per l'italiano in LaTeX», *ArsTeXnica*, 1, http://www.guit.sssup.it/arstexnica/download_ars/arstexnica01.pdf. (Citato a p. 149.)

Eco, Umberto

1977 *Come si fa una tesi di laurea. Le materie umanistiche*, Bompiani, Milano. (Citato alle pp. 116, 119.)

Fairbairns, Robin

2011 *The UK TeX FAQ*, http://www.tex.ac.uk/tex-archive/help/uk-tex-faq/newfaq.pdf. (Citato alle pp. 31, 60.)

Flynn, Peter

2005 «In my opinion», *PracTeX Journal*, 1, http://tug.org/pracjourn/2005-1/flynn/. (Citato a p. 7.)

Goossens, Michel, Frank Mittelbach e Johannes Braams

2004 *The LaTeX Companion*, Addison-Wesley, Reading (Massachusetts).

Gregorio, Enrico

2009 *Appunti di programmazione in LaTeX e TeX*, http://profs.sci.univr.it/~gregorio/introtex.pdf. (Citato a p. 128.)

2010 *LaTeX. Breve guida ai pacchetti di uso più comune*, http://profs.sci.univr.it/~gregorio/breveguida.pdf. (Citato alle pp. 20, 101, 104, 140.)

2011 *Installare TeXLive 2010 su Ubuntu*, http://profs.sci.univr.it/~gregorio/texlive-ubuntu.pdf. (Citato a p. 14.)

Guiggiani, Massimo e Lapo Filippo Mori

2008 «Consigli su come *non* maltrattare le formule matematiche», *ArsTeXnica*, 5, http://www.guit.sssup.it/arstexnica/download_ars/arstexnica05.pdf. (Citato a p. 61.)

Knuth, Donald Ervin

1973 *Computer Programming as an Art*, Addison-Wesley, Reading (Massachusetts), vol. 3.

1984 *The TeXbook*, Addison-Wesley, Reading (Massachusetts).

Lamport, Leslie

1994 *LaTeX. A Document Preparation System*, Addison-Wesley, Reading (Massachusetts).

Lesina, Roberto

2004 *Il Nuovo Manuale di Stile*, edizione 2.0, con la collaborazione di Federico Boggio Merlo, Zanichelli, Bologna. (Citato alle pp. 149, 153.)

Mittelbach, Frank, Gianluca Pignalberi e Dave Walden

2007 «Intervista a Frank Mittelbach», *ArsTeXnica*, 3, http://www.guit.sssup.it/arstexnica/download_ars/arstexnica03.pdf. (Citato a p. 6.)

Mori, Lapo Filippo

2006 «Tabelle su LaTeX 2ε: pacchetti e metodi da utilizzare», *ArsTeXnica*, 2, http://www.guit.sssup.it/arstexnica/download_ars/arstexnica02.pdf. (Citato a p. 90.)

2007 «Scrivere la tesi di laurea con LaTeX 2ε», *ArsTeXnica*, 3, http://www.guit.sssup.it/arstexnica/download_ars/arstexnica03.pdf. (Citato alle pp. 34, 105.)

Mori, Lapo Filippo e Maurizio Himmelmann

2007 «Scrivere il curriculum vitae», *ArsTeXnica*, 4, http://www.guit.sssup.it/arstexnica/download_ars/arstexnica04.pdf.

Oetiker, Tobias *et al.*

2011 *The Not So Short Introduction to LaTeX 2ε*, http://www.ctan.org/tex-archive/info/lshort/english/lshort.pdf. (Citato alle pp. 6, 43.)

Pakin, Scott

2009 *The Comprehensive LaTeX Symbol List*, http://www.ctan.org/tex-archive/info/symbols/comprehensive/symbols-a4.pdf. (Citato alle pp. 50, 74.)

Pantieri, Lorenzo

2007 «Introduzione allo stile ClassicThesis», *ArsTeXnica*, 6, http://www.guit.sssup.it/arstexnica/download_ars/arstexnica06.pdf. (Citato alle pp. 28, 30, 51.)

2009 «L'arte di gestire la bibliografia con biblatex», *ArsTeXnica*, 8, http://www.guit.sssup.it/arstexnica/download_ars/arstexnica08.pdf.

Pantieri, Lorenzo e Tommaso Gordini

2009 *L'arte di fare una presentazione con* beamer, http://www.lorenzopantieri.net/LaTeX_files/Presentazioni.pdf. (Citato a p. 167.)

Sabatini, Francesco e Vittorio Coletti

1997 *il Sabatini Coletti. Dizionario della Lingua Italiana*, Giunti Gruppo Editoriale, Firenze.

Serianni, Luca

1989 *Grammatica italiana. Italiano comune e lingua letteraria*, con la collaborazione di Alberto Castelvecchi, UTET, Torino.

INDICE ANALITICO

D

E

F

G

H

I